

THÈSE de DOCTORAT DE L'ÉCOLE CENTRALE DE LYON
opérée au sein de l'Université de LyonÉcole Doctorale n°160
Électronique Électrotechnique et Automatique (EEA)

Discipline : Électronique, Micro et Nanoélectronique, Optique et Laser

Soutenue publiquement le 5 juillet 2023, par :

Mayeul Cantan

Energy-Efficient Computing with
Integrated Ferroelectrics for Embedded
and Edge Devices

Devant le jury composé de :

Maneux, Cristell	Professeur des Universités ^a	Université de Bordeaux / IMS Bordeaux	Présidente
Niemier, Michael	Full Professor	University of Notre Dame, Indiana, USA	Rapporteur
Portal, Jean-Michel	Professeur des Universités ^a	Aix-Marseille Université / IM2NP	Rapporteur
O'Connor, Ian	Professeur des Universités ^a	École Centrale de Lyon / INL	Directeur de thèse
Deleruyelle, Damien	Professeur des Universités ^a	INSA Lyon / INL	Co-directeur de thèse
Marchand, Cédric	Maître de Conférences ^a	École Centrale de Lyon / INL	Co-directeur de thèse
Slesazek, Stefan	Senior Researcher	NaMLab, Dresden, Allemagne	Invité
Giraud, Bastien	Ingénieur Chercheur	CEA-LIST	Invité

^a63^{ème} section

UNIVERSITÉ DE LYON

Abstract

École Centrale de Lyon
Department Name

PhD

Energy-Efficient Computing with Integrated Ferroelectrics for Embedded and Edge Devices

by Mayeul CANTAN

Ferroelectric materials are gaining traction in integrated circuits, notably thanks to the introduction of Hafnium-Zirconium Oxides, which are compatible with state-of-the-art semiconductor materials and manufacturing technologies. Their ferroelectric properties, combined with regular **CMOS** technology, enable interesting new circuit architectures. Bringing **Non-Volatile Memory** technology closer to compute elements unlocks opportunities for higher power efficiency by reducing data transfers, lowering static power consumption, and enabling **normally-off** computing schemes.

In this thesis, ferroelectric materials are approached from a circuit design perspective, providing a basic understanding of their properties and modeling approaches. Multiple circuit architectures enabled by ferroelectric materials are also presented, both with **Back-End of Line** and **Ferroelectric Field-Effect Transistor (FeFET)** technologies, and electrical characterization results where available. Finally, projected performance figures are extracted, to allow comparison with more mature technologies, both at the circuit and system level, through **Design-Space Exploration (DSE)** techniques and a custom system-level performance evaluation framework.

Obtained results include novel circuit designs, some of which were fabricated with 130 nm and 28 nm technology, **DSE** simulation results for metrics such as memory window and energy consumption, as well as multiple software tools created during the project.

Acknowledgments

This document may bear my name on the first page, but I wouldn't have been able to complete it without the direct and indirect contributions of the people who helped me along this journey, and those that encouraged me. I could not include everyone on this page, but to those whose name isn't written down: thank you for your help and support over the course of my PhD.

I am first and foremost grateful to my supervisor Ian O'Connor, who accompanied me during this thesis, providing support and advice, and being responsive as I faced difficulties. My co-supervisors Cédric Marchand and Damien Deleruyelle were of great help for discussing simulations and physics of ferroelectric devices; and for the review of this document, as well as Damien's more direct contributions to [subsection 2.2.2](#).

My thanks extend to my esteemed colleagues that I also call friends for their support over the years: Clément Zrounba and Arnaud Poittevin helped me with the layout of [MAD200](#) designs discussed in [chapter 3](#), and partook in numerous fruitful scientific discussions. Etienne Dupuis, for his contagious motivation and his open heart, Adil Brik, Lucien Del Bosque, followed by the next generations of PhD students were always ready to help, whether it was work-related or with personal matters. This resulted in a welcoming work environment, where I made some of my fondest memories.

For their help and support, I would like to thank the rest of my team members at [INL](#), including Alberto Bosio, for his help with design synthesis in the context of the image filter presented in [section 4.6](#). Other [3εFERRO](#) project members were also of great help, including at [INL](#), where Jordan Bouaziz and other members of the Materials team, including Ingrid Cañero Infante, Bertrand Vilquin and Pedro Rojo Romeo, patiently taught me over and over the physics and fabrication aspects of ferroelectric materials. I also owe some of my understanding of ferroelectrics to [NaMLab](#) personnel, especially Stefan Slesazek and Evelyn Breyer, who carried out much of the design and implementation work of the convolutional image filter discussed in [section 4.6](#). Evelyn was instrumental to the work presented in this document by providing the "Preisach" ferroelectric simulation model used for most simulations, though the first modeling discussions and experimental data I obtained were courtesy of Carlotta Gastaldi from [EPFL](#), which enabled my first "Landau"-based simulations.

For successively helping me develop the system-level benchmarking platform described in [section 5.4](#), I would like to thank in chronological order master students Pierre-Etienne Polet and Luca Mozzone, and postdoctoral researcher Marcello Traiolla. They successively provided the main development effort for implementing my designs, and provided great feedback on them. Their documentation efforts were tremendously helpful while writing this section.

For their support, and constant encouragements, I would also like to thank my family, including my parents and siblings, who constantly teased me about the progress on my manuscript, while doing their best to support me during these trying times. Writing this manuscript was not an easy task, neither for me nor the people surrounding me, so I am grateful to Liz, for her patience, understanding and support.

I found unconditional support among my friends as well, who I have not forgotten despite seeing them less often, and whom I miss after all that time spent working isolated. You will recognize yourselves, and I hope to see you again soon! I particularly want to thank Claire Segovia for taking some of her time to help me work through the early stages of writing this manuscript, and helping me overcome writer's block and reject distractions, which was extremely helpful.

Less conventionally, I would also like to thank Romano Giannetti, the maintainer of the circuitikz circuit illustration library that I have used extensively throughout this document, for answering my request to create ferroelectric capacitors and transistors symbols¹. These thanks extend to the maintainers of the countless open source tools used over the course of this PhD, as well as Laurent Carrel for administrating our local computing resources.

Lastly, I am grateful to the European Commission for funding interesting and relevant projects, which enabled me to conduct this work as part of the [3εFERRO](#) project that received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°780302.

¹<https://github.com/circuitikz/circuitikz/issues/515>

Contents

Abstract	3
Acknowledgments	5
1 Introduction	15
1.1 About this document	15
1.1.1 License	15
1.1.2 Internal links and color code	15
1.1.3 Document source, errata and supplementary material	16
1.1.4 Aim of this document	16
1.2 Context	16
1.2.1 Internet of Things and edge computing	16
Internet of Things	16
Edge computing	17
1.2.2 The demise of Dennard scaling and Moore's law	17
Moore's Law	17
Dennard scaling	17
"Happy scaling" consequences	18
End of Dennard scaling	18
End of Moore's Law	18
1.2.3 Von Neumann Architecture	19
1.2.4 Von Neumann Bottleneck	19
1.2.5 Ferroelectric HfZrO ₂	20
1.2.6 Conclusion	21
1.3 3εFERRO European project	21
1.3.1 Project partners	21
1.3.2 Project Goals	22
Achievements	22
Contributions	22
2 Ferroelectrics: behavior and modeling	23
2.1 Ferroelectricity	23
2.1.1 Ferroelectric crystals	23
Coercive field	24
Polycrystals and domains	24
Note on the nomenclature used for write and erase operations	27
2.1.2 <i>P-V</i> curve	27
2.1.3 Relationship with capacitance and paraelectricity	27
Charge screening and depolarization field	29
Ferroelectric Tunnel Junction	30
2.1.4 PUND measurements	30
2.2 Modeling	34
2.2.1 Landau model	34
Description	34
Expression	35
Usage	36
Fitting	36
Conclusion	37
2.2.2 Preisach model	37

	Acknowledgements	37
	Hysteron	37
	Cumulative hysteron behavior	40
	Limitations	40
2.2.3	Simplified model for large-scale simulation	43
2.3	Ferroelectric capacitors	44
2.3.1	Regular capacitor	44
2.3.2	Non-volatility	44
2.3.3	Negative capacitance	45
2.4	Ferroelectric transistors	45
2.4.1	FeFET Devices	45
	Drawbacks	46
2.4.2	Gate stacks	46
2.4.3	Modeling	47
2.5	State-of-the-art on ferroelectric circuits	47
2.5.1	Ferroelectric Hafnia	47
	Modeling	48
2.5.2	Ferroelectric Capacitors and Back-End of Line circuit design	48
2.5.3	Ferroelectric Field-Effect Transistors-based circuit design	48
	P-channel Ferroelectric Field-Effect Transistors	49
2.5.4	Comparison with other Non-Volatile Memories	49
2.5.5	Design-Space Exploration	49
2.5.6	System-level performance evaluation	51
	Hardware prototyping	51
	FPGA-based emulators	52
	Software simulators	52
	Compiler support and code instrumentation	52
3	Ferroelectric capacitors-based designs	53
3.1	Introduction	53
3.1.1	Back-End of Line technology	53
3.1.2	MAD200 process	54
3.2	1T1C memory bitcell	54
3.2.1	Operation	55
	Bitcell selection and programming	55
	Bitcell readout	57
	Multi-level memory	57
3.2.2	Simulation	58
	MAD200-based simulations	58
3.3	FeFET-like structure	58
3.3.1	Description	58
3.3.2	Design	59
	Capacitance matching	60
3.3.3	Characterization	62
	Protocol and results	62
3.3.4	Extension to multi-transistor circuits	64
3.4	Destructive-read TCAM	65
3.4.1	Description	65
	Ternary Content-Addressable Memory	65
	Operating principle	66
	Limitations	67
3.4.2	Design	67
3.5	2T1C versatile bitcell	68
3.5.1	Description	68
	Programming	69
	Reading as FTJ – Non-destructive read	69
	Reading as DRAM, or 1T1C – Destructive read	70
	Reading as FeFET – Non-destructive read	70

	2T-nC	70
	Destructive TCAM emulation	71
3.5.2	Design	71
	Capacitance matching for DRAM operation with destructive read	71
	2T-nC	72
3.5.3	Characterization results	74
	Reference I_{DS} — V_{GS} plot for Q_R	74
	Characterization protocol	74
	Results and interpretation	74
	Switching dynamics investigation	75
3.6	Conclusion	76
3.6.1	1T1C memory bitcell	76
3.6.2	Back-End of Line FeFET-like structure	76
3.6.3	Destructive-read TCAM	77
3.6.4	2T1C	77
4	Ferroelectric transistor-based designs	79
4.1	Introduction to FeFET circuits	79
4.1.1	Programming the ferroelectric oxide	80
4.1.2	V_{th} shifting	80
	Analog V_{th} shift control	80
	Dual-state V_{th} shift control	82
4.1.3	Comparison with CMOS-based logic	82
	Advantages compared to CMOS-based logic	82
	Deficiencies compared to CMOS-based logic	83
	Technology process and p-FeFET availability	83
4.2	1T-FeFET memory	83
4.2.1	Operating principle	84
	Read operation	84
	Write operation	84
4.2.2	Comparison with other floating-gate transistor memories	85
4.2.3	Possible hybrid operation mode	85
4.3	Transresistance circuits	86
4.3.1	Complementary logic with p-FeFET	86
4.3.2	Resistive logic	87
4.3.3	Dynamic logic	87
	Hybrid dynamic logic with CMOS stages	88
4.3.4	Pass-transistor logic	89
	FeFET-based pass-transistors	89
4.4	Non-volatile FeFET-based logic gates	89
4.4.1	NV-NAND2	89
4.4.2	NV-AND2	90
4.4.3	NV-XOR2	90
4.5	FeFETs as add-on technology	91
4.5.1	Black & Das memory cell as a checkpointing mechanism	91
4.6	Convolutional Image Filter with FeFET-based Logic-in-Memory	92
4.6.1	Choice of a convolutional image filter	92
	Convolution operation in one dimension	93
	Convolution of a two-dimensional image	94
	Required post-processing	94
4.6.2	Filter architecture	95
	Intermediate samples	95
	Alternate scan-chain	97
	Bit precision	97
4.6.3	FeFET-based Logic-in-Memory multiplier design	98
	Multiplier circuit, adder circuit	98
	Ripple-Carry Adder	98
	Pipelined architecture	98

	Kernel weight programming	99
4.6.4	Validation in simulation and identified issues	103
	Simplification of the circuit-level simulation	103
	Generation of reference input and output signals	104
	Wrong clock trigger for multiplier	105
	Node discharge dependency	106
4.6.5	Results	106
	Weight programming	106
	Operating point	107
	Complete interactive demonstrator	108
4.7	Conclusion	108
4.7.1	FeFET-based logic	108
4.7.2	Image filter	110
4.7.3	FeFET-based memories	110
5	Design space exploration and optimization	113
5.1	Introduction to design space exploration	114
5.1.1	Parameter space and performance space, Pareto optimal	114
	Parameter and performance space	114
	Multi-objective optimization	115
	Pareto front	115
5.1.2	Tool-assisted exploration	115
	Process	116
5.1.3	System-level benchmarking	117
5.2	Design space exploration tools	117
5.2.1	LIFT optimizer	117
5.2.2	Cadence IPC	118
5.3	Design space exploration results	119
5.3.1	Sampling of 1T1C bitcell design space	119
	Problem description and expected results	119
	Test circuit	119
	Problem definition	120
	Metrics extraction	123
	Results	123
5.3.2	Non-volatile FeFET-based NAND gate (NV-NAND2)	125
	Test circuit and parameter space	125
	Performance space	127
	Results	127
5.4	System-level benchmarking Platform	129
5.4.1	Introduction	129
	Objectives	129
	Use-cases	129
	Acknowledgements	129
5.4.2	Scope of the benchmarking platform	129
	Target System Architectures	129
	Extracted Metrics	129
	Benchmarks	130
5.4.3	Implementation	130
	Architecture	131
	Simulation pipeline	132
	Decoder Control Module	132
	Output Manager Control Module	132
	Computation and performance tracking	132
5.4.4	Operation modules and model cards	133
	Memory module	133
	Operation modules	133
	Operation module implementation	133
	Model card structure	134

5.4.5	Example case: Adder	134
5.5	System-level exploration results	135
5.5.1	Normally-off use-cases	136
5.5.2	Interpolator simulations	137
5.5.3	Matrix multiplication benchmark	138
5.6	Conclusion	139
5.6.1	Design-Space Exploration	139
	Model instability and accuracy	139
5.6.2	System-level benchmarking platform	140
	Current status	140
	Further refinements and analysis	140
	Design-Technology Co-Optimization	140
6	Conclusion	143
6.1	Back-End of Line ferroelectric technology	143
6.2	Current FeFET strengths and limitations	144
6.2.1	Future of FeFET technology	144
6.3	Automated DSE and modeling	144
6.3.1	Modeling issues	144
6.4	System-level performance evaluation	145
6.5	Short-term perspectives	145
6.5.1	Remaining characterization work	145
6.5.2	Future simulations	145
6.6	Future considerations for ferroelectric technology	146
6.6.1	Space efficiency	146
6.6.2	Control signals	146
	Bibliography	147
	Glossary	159
	Acronyms	163
A	Code listings	165
B	Additional circuits	181
C	Additional tables	183

List of Figures

1.1	Von Neumann Architecture diagram	20
2.1	Effect of domain orientation on external and coercive field mismatch.	25
2.2	Ferroelectricity illustration with spring-charge equivalent system	25
2.3	Experimental domain orientation measurement in Si-Doped HfO ₂	26
2.4	Reading P_r and V_C on a curve	28
2.5	Example P - V curves	28
2.6	Electrical polarization and relationship with ferroelectricity.	29
2.7	Band diagrams with screening length	31
2.8	PUND measurement illustration	32
2.9	Pure ferroelectric P - V response extracted through PUND stimulation	34
2.10	Plots of Gibbs free energy and its derivative for the Landau model	35
2.11	Fitting a Landau polynomial to experimental data	38
2.12	Example ferroelectric hysteresis cycle	39
2.13	Preisach model – hysterons and plane	39
2.14	(V_c^+, V_c^-) 2D Gaussian distribution	41
2.15	Internal loops and turning points	42
2.16	Example of 2D V_c^+/V_c^- Gaussian distribution extraction	42
2.17	Handling arbitrary V_c^+/V_c^- distributions	42
2.18	FeFET symbol and gate stack	45
2.19	FeFET gate stacks	47
2.20	System-level performance evaluation landscape	51
3.1	Illustration of front-end and back-end ferroelectric technologies	54
3.2	Electron microscope image of the MAD200 layers	55
3.3	Cutaway comparison of a FeFET and PsFeFET stack	59
3.4	Comparison cutaway of FEOl and BEOl ferroelectric technologies	60
3.5	3D PsFeFET layout	63
3.6	$I_D = f(V_G)$ characteristic of a PsFeFET	64
3.7	Programming voltage and 2T1C memory window	75
3.8	Pulse width effect on 2T1C memory window	76
4.1	V_{th} shifting illustration	81
4.2	Black & Das cell behavior	92
4.3	Filter kernel examples	93
4.4	Illustration of a convolutional image filter	94
4.5	High-level diagram of the proposed image filter.	95
4.6	Input and output signals of the programming voltage multiplexer	103
4.7	Verification flow employed to validate the image filter design	104
4.8	Measurements of the V_{th} shift depending on pulse duration and voltage	107
4.9	Image filter dynamic characterization results and operating point	109
5.1	Illustration of complexity increase due to combinatorial explosion.	114
5.2	Parameter space and performance space; Pareto set and front	115
5.3	Experimental and Simulation-based approaches to performance measurement	116
5.4	Automated Design-Space Exploration approach	117
5.5	Pareto front and set generation flow	117
5.6	Cadence IPC architecture	118
5.7	Annotated 1T1C test waveforms	121

5.8	DSE results for 1T1C exploration: time, memory window vs transistor geometry	124
5.9	DSE results for 1T1C exploration: time and energy vs capacitor area	126
5.10	Preliminary DSE of the non-volatile FeFET-based NAND gate	128
5.11	Illustration of coarse- and fine-grained LiM	130
5.12	Illustration of a Logic-in-Memory-based accelerator on the system bus	130
5.13	Benchmarking platform internal structure diagram	131
5.14	Simulation platform architecture: model cards and operation manager	133
5.15	Illustration of the common operation interface	133
5.16	Execution diagram of the operation module	134
5.17	Addition operation performed by the example accelerator	135
5.18	Benchmarking platform inputs and outputs	135
5.19	Benchmarking platform results for normally-off computing	137
5.20	Exploration of LiM trade-off: interpolator vs LUT multiplier	138
5.21	Energy use of WB and NWB strategies for matrix multiplication	139
5.22	DTCO pipeline combining system-level performance and device-level parameters	141

List of Circuits

3.1	1T1C bitcell	55
3.2	4×4 1T1C array	56
3.3	PsFeFET and FeFET electrical circuits	59
3.4	PsFeFET and equivalent schematic	61
3.5	Equivalent PsFeFET floating capacitors circuit	61
3.6	PsFeFET layout	63
3.7	PsFeFET-based CMOS with shared FeCap	65
3.8	Destructive-read TCAM schematic	66
3.9	Destructive-read TCAM layout	68
3.10	2T1C bitcell schematic	69
3.11	2TnC schematic	70
3.12	2T1C layout	72
3.13	2T3C structure as designed	73
4.1	FeFET Symbol	80
4.2	1T-FeFET bitcell	83
4.3	4×4 1T-FeFET array	84
4.4	Transresistance circuit with complementary FeFET	86
4.5	FeFET circuit with resistor-based transresistance	87
4.6	Transresistance with dynamic logic architecture	88
4.7	Hybrid CMOS and dynamic logic architecture	88
4.8	FeFET-based pass transistor and associated truth table	89
4.9	NAND FeFET-based logic gate	90
4.10	AND FeFET-based logic gate	90
4.11	XOR FeFET-based logic gate	90
4.12	Black & Das FeFET-enhanced SRAM	91
4.13	Image filter architecture diagram.	96
4.14	Shift register internal diagram	97
4.15	Final FeFET-based Full-Adder multiplier.	99
4.16	Ripple-Carry Adder based on full adders	100
4.17	Image filter multiplier pipeline stages	101
4.18	Voltage multiplexer used for programming the multiplier FeFET	102
5.1	1T1C bitcell schematic and test power supplies	120
5.2	Dynamic NV-NAND2 circuit for DSE	127
B.1	First version of the full adder for the multiplier circuit	181
B.2	Second iteration of the full adder for the multiplier circuit	182
B.3	Cadence circuit for 1T1C test bench	182

Listings

2.1	Verilog-A Minimal ferroelectric model	43
5.1	Memory model card template	134
5.2	Operation model card template	134
5.3	Execution trace for performing an addition	135
A.1	SKILL® Metric extraction from waveforms for 1T1C bitcell	165
A.2	SKILL Metric extraction from waveforms for FeFET-based NAND gate . . .	167
A.3	1T1C Design space exploration python script	169
A.4	FeFET-based non volatile NAND design space exploration script	170
A.5	Verilog-A Data Serializer	171
A.6	Verilog-A Data Serializer with enable signal	172
A.7	GNU Octave code for fitting Landau coefficient to experimental curves . . .	173
A.8	GNU Octave code for orthogonal projection (dependency)	176
A.9	GNU Octave code for polynomial fit with constraints on coefficients (dependency)	176
A.10	GNU Octave code for image filtering with convolutionnal kernels	177
A.11	Verilog description of falling edge sensitive shift register with alternate input	177
A.12	Verilog description of shift register with alternate input and adjusted sampling	178
A.13	Synthesized Verilog description of 1-bit shift register	178
A.14	Kernel programming FeFET-based image filter Verilog testbench	179

Chapter 1

Introduction

Contents

1.1 About this document	15
1.1.1 License	15
1.1.2 Internal links and color code	15
1.1.3 Document source, errata and supplementary material	16
1.1.4 Aim of this document	16
1.2 Context	16
1.2.1 Internet of Things and edge computing	16
1.2.2 The demise of Dennard scaling and Moore's law	17
1.2.3 Von Neumann Architecture	19
1.2.4 Von Neumann Bottleneck	19
1.2.5 Ferroelectric HfZrO ₂	20
1.2.6 Conclusion	21
1.3 3ϵFERRO European project	21
1.3.1 Project partners	21
1.3.2 Project Goals	22

1.1 About this document

1.1.1 License

This work is licensed under a [Creative Commons “Attribution 4.0 International”](https://creativecommons.org/licenses/by/4.0/deed.en) license.



In simple terms, this means you are free to use both the text (including code) and images of this document, as well as remixing and adapting it, as long as you properly attribute it. For more details, please consult the above link, or the Creative Commons website at <https://creativecommons.org/licenses/by/4.0/deed.en>. The license does not apply to figures provided by external sources, as noted in the captions.

1.1.2 Internal links and color code

This document contains both internal and external links that can be followed by a compatible reader. These links are color-coded as follows:

1. **Internal links** to sections, figures, code listings and various parts of the document
2. **Internal links** to the **Bibliography** section
3. **Internal links** to the **Glossary** and list of **Acronyms**
4. **External links** to websites

1.1.3 Document source, errata and supplementary material

An up-to-date revision of this document, as well as its L^AT_EX source code and supplementary material (code listings as well as additional data) are available at <https://thesis.mayeul.net>¹.

1.1.4 Aim of this document

This document aims to both summarize the work accomplished and the results obtained over the course of this PhD; as well as serving as a self-contained introductory document to ferroelectric circuits design.

1.2 Context

For decades, the microelectronics industry has seen exceptional growth in both size and capabilities. Constant technological improvements were achieved through miniaturization, bringing both continuous performance and efficiency gains. The industry is now structured around these expected gains, but while there initially was “plenty of room at the bottom” (to cite Feynman) for miniaturization, this is not the case anymore, and continuing to increase the density of integrated circuits will entail geometry changes, as **Fin FETs** (**FinFETs**) are now common in most advanced nodes, and vertical nanowires[Poi22] are envisioned.

Another possible avenue for performance and energy efficiency increases does not involve increasing density, but rather enable new functionality at a similar miniaturization level. This is the case for changes to devices such as **Fully Depleted Silicon on Insulator** (**FDSOI**) transistors and **Ferroelectric Field-Effect Transistors**, but also for changes to computing architectures, that can provide tangible performance gains.

Besides economic considerations (new technology driving sales by obsoleting previous generations of devices), energy efficiency gains are important to offset the growing ecological and energetic footprint of electronics devices. These gains are offset to an extent by a rebounding effect, as cheaper, more powerful and more energy efficient devices enable new use-cases. This however brings an increase in capabilities that is hardly quantifiable, and makes technology affordable.

In this introduction, the existing trends in microelectronics are summarized, highlighting why the previous approach to efficiency gains is not sufficient anymore. New use-cases such as the **Internet of Things** (**IoT**) and **edge computing** are described, as well as one specific emerging technology, based on ferroelectric **HfZrO₂**, and the associated **3eFERRO** European project this work is part of.

1.2.1 Internet of Things and edge computing

Internet of Things

The **IoT** comprises a multitude of devices and device classes. Those are usually sensors, that measure physical quantities such as temperature, humidity or atmospheric pressure, or more specific values such as energy usage, the number of cars in a parking lot, movement sensors as part of an alarm system, etc. More complex devices can also be considered **IoT**, such as modern, Internet-connected cars, or “smart” devices that are getting more common in homes, like thermostats, network-enabled kettles, lights, curtains, cat litters... These devices sometimes include actuators as well as sensors, some of which can be safety-critical, such as a barrier controlling access to a highway emergency exit ramp.

IoT is a fast-growing market, as data collection, and access to supplementary data from the Internet enables much more complex behaviors. For instance, the use of temperature and soil humidity sensors could help monitor crops; and connecting an irrigation system to weather forecasts could save water. However, as opposed to the **cloud**, which is made up of expensive, powerful servers in electrical grid-connected datacenters, **IoT** is usually cheap, expendable, with energy and connectivity constraints. Cost and power-efficiency means

¹mirrors available at <https://these.mayeul.net>, <https://mayeulc.gitlab.io/thesis>, <https://thesis.mayeul.cantan.eu> and <https://mayeulc.github.io/thesis>

that **IoT** devices generally have very limited amounts of memory and computing power. Indeed, be it a moisture sensor in the middle of a field, a pressure sensor on the rim of a car wheel, or an open/closed sensor at the top of a window, supplying electrical power to these devices is often challenging, leading to a large number of battery-operated devices, sometimes supplemented by energy harvesting mechanisms: solar panels, triboelectricity or vibrational energy harvesters, peltier or thermoelectric modules to harvest waste heat, etc. This also means these devices often rely on a wireless connection to transmit data, which requires a relatively important amount of power: according to the datasheet[Esp22], a very common ESP8266EX typically consumes around 50 mW with its radio modem turned off, while transmitting data at 15 dBm uses an order of magnitude more power. With its CPU turned off, that device only consumes only 0.5 mW to 20 mW.

Edge computing

IoT makes extensive use of the **cloud** as a data source and sink. However, as detailed above, transmitting data has a non-negligible energy cost for battery-backed devices. This also becomes a bandwidth issue once data reaches the servers: Internet transit costs can be quite substantial, and processing cost rises with the amount of data.

Edge computing aims to relocate some of the processing tasks outside of the **cloud**, to its periphery or *edge*. This concerns data processing that is too intensive to perform on the **IoT** devices themselves due to resource constraints, but where applying some amount of pre-processing can reduce latency and bandwidth sent to the **cloud**.

In practice, processing data as close as possible to the source usually results in efficiency gains. For instance, in a video surveillance system, a wireless camera (**IoT** device) can avoid transmitting data when no movement is detected, and compress the video feed before sending it to save energy during transmission. That feed would then be sent to a more powerful intermediate *edge device* for further processing. This could include an image recognition task, identifying a fox in a chicken farm, or an armed person in a crowd. That edge device therefore performs relatively intensive processing tasks to pre-filter data before sending it to remote servers for further processing. In the above example, this could be archiving the video, or transmitting it to another device.

Another example is voice assistant can also perform basic voice recognition locally, to answer commands faster, but also to send text instead of voice to the **cloud**; this also has availability and privacy benefits.

1.2.2 The demise of Dennard scaling and Moore’s law

Moore’s Law

Moore’s law was formulated in 1965[Moo65] by Gordon Moore (Intel co-founder, 1929–2023), stating that the number of features (i.e. transistors) on an integrated circuit doubled every two years for the best-priced chips (smallest cost per feature). This is usually roughly summarized as transistor density doubling every two years. That observation held for much longer than initially anticipated, although that is partly due to the industry using it as a roadmap for development, including factory investments, R&D spending, and product planning.

Dennard scaling

After the industry’s transition to **Metal-Oxide-Semiconductor Field-Effect Transistor (MOS-FET)** transistors, Dennard scaling was identified in 1974[Den+74] as a practical way to continue increasing the transistor density, by reducing transistor dimensions as specified in **Table 1.1**. The values are constructed by scaling transistors down while keeping the electrical field constant.

This scaling method led to a sustained density improvements over the following years, the period from 1980 to 1995 being nicknamed the era of “happy scaling”. Thanks to improved manufacturing methods, devices and circuits could be scaled down at a regular pace. The reduced component and interconnection size lowered capacitance, increasing both maximum operating frequency, and energy efficiency by reducing switching losses.

Device or Circuit Parameter	Scaling Factor
Device dimension t_{ox}, L, W	$1/\kappa$
Doping concentration N_a	κ
Voltage V	$1/\kappa$
Current I	$1/\kappa$
Delay time per circuit $V \cdot C/I$	$1/\kappa$
Power dissipation per circuit $V \cdot I$	$1/\kappa^2$
Power density $V \cdot I/A$	1

TABLE 1.1: Dennard scaling parameters and results on performance, from [Den+74]. Note the constant power density.

To remain in line with Moore’s law, κ was chosen to halve transistor surface area $A \approx L \cdot W$ each generation: $1/\kappa^2 = 1/2 \implies \kappa = \sqrt{2} \approx 1.4$ [Boh07], roughly every two years.

“Happy scaling” consequences

Constant performance gains had a rippling effect across the industry: it was often possible to reuse previous chip designs, scale them down, manufacture them with the improved process, and obtain performance gains as a result. Moreover, algorithms designed for general-purpose processors could reap these benefits as well, provided backwards compatibility was maintained across chip generations. Maintaining backwards compatibility with older hardware and software generations proved beneficial for iterative improvements, and lasting investments.

As a result, general purpose processor architectures, and associated algorithms, toolchains and compilers have leapfrogged specific-purpose hardware: in some cases, ASICs and custom processor architectures may have offered performance and efficiency benefits over Central Processing Unit (CPU)-based implementations, but the long design process combined with steady performance increases means that CPU performance could catch up with the application-specific design before it was available.

The current hardware and software ecosystem must be observed through that lens, and this work reflects the recent interest renewal in alternative approaches.

End of Dennard scaling

Dennard scaling is not sustainable indefinitely, as there are engineering, manufacturing and physical limits to doing so: voltages cannot drop below a certain level, as this would degrade the signal-to-subthreshold current noise ratio. Doping concentrations becomes hard to increase as size shrinks, leading to issues such as uneven atomic distribution and direct band-to-band tunneling across PN (transistor source and drain) junctions [Boh07]. Dimensions also have fundamental limits, as tunneling current becomes non-negligible below a certain size (depending on the voltage and other factors, around 20 nm).

As feature size shrunk below 65 nm, leakage currents became more important, starting with transistor gate insulators, which were 1.2 nm, or about five SiO₂ atomic layers thick by 2005 [Boh07]. This led to the introduction of “high-k” dielectrics in order to maintain the same electrical field in the transistor channel despite using thicker gate oxides, reducing leakage current, but also breaking with Dennard scaling. Beyond this point, constant-field scaling encountered more issues which required rethinking the approach to device scaling.

The end of Dennard scaling had a notable effect on integrated circuits power dissipation: as voltages could not be lowered much further, and leakage currents were increasing, the power density could not remain constant anymore. This resulted in substantial power usage increases for modern chips and processor circuits. Power efficiency can still be improved thanks to increased performance, while leakage current can be mitigated by disabling the power supply of unused chip areas, a so-called “dark silicon” approach.

End of Moore’s Law

While transistors were able to continue scaling down post-Dennard scaling for a few years, the end of planar MOSFET technology scaling was in sight. With no clear scaling roadmap,

investments required to follow the roadmap plotted by Moore continued to grow rapidly, further consolidating the semiconductor industry.

More recently, density increases fell short of Moore’s predictions, and have been achieved through more important changes to the **Field Effect Transistor (FET)** architecture, with the generalization of **FinFET**, and the development of **Gate-All-Around FET (GAAFET)** and vertical nanowire **FETs**. These designs achieve transistor density gains largely by exploiting the vertical dimension, without substantially reducing feature size.

Another avenue for continuing to increase the number of features per chip is to increase chip size, with recent developments being made in wafer-scale chips. However, this is unlikely to completely offset the rising costs of the latest generation semiconductor nodes, that reflect the important investments necessary to continue increasing transistor density.

Future performance improvements are expected to come from both density increase (albeit at a slower pace), but also from new system architectures and devices, possibly non-electrical (photonics, molecular storage). A variety of labels have been used to describe these approaches, including “more Moore” and “more than Moore”.

Power efficiency is possibly an even greater challenge than continued density improvements, without constant power-density Dennard scaling. Thermal issues are worsened by the move away from planar transistors, which were able to maximize contact area with the silicon die for thermal dissipation.

Reflecting this change of direction, the industry reinvents itself, and is investigating multiple emerging technologies, including a re-evaluation of some that fell out of favor during the “happy scaling” era. The reduced pace of performance improvement for conventional transistor may also finally allow alternatives to catch up performance-wise. This work presents one such thread of investigation into ferroelectric materials for bringing new functionalities to existing semiconductor technologies.

The **International Technology Roadmap for Semiconductors (ITRS)**, coordinating different companies toward the development of next-generation silicon devices also illustrated this trend in 2016, becoming the **International Roadmap for Devices and Systems (IRDS)**. Their reports offer an overview of the technologies under development and their potential applications[[IRDS22](#)].

1.2.3 Von Neumann Architecture

Computers today are overwhelmingly designed around the concept of stored programs: by storing programs in memory, these can be easily loaded and modified, as well as copied and transferred. This makes computers much more versatile by not requiring to physically alter them in order to perform a new function[[Knu70](#)], unlike early architectures.

One of the most used computer architectures following this paradigm is the Von Neumann architecture pictured in [Figure 1.1](#), and its derivatives[[Paw22](#)]. Named after John von Neumann, it reads both instructions and data from the same memory, through a memory bus. As a generalization of this architecture, multiple peripherals, including **Input/Output (I/O)** devices, can be present on the memory bus (or system bus) at known addresses, and communicate using the same interface made of a data and address bus, with read and write operations. This generalization helped make computer architectures very modular, only having to rely on a common abstraction for most use-cases. “Peripherals” such as additional **I/O** adapters, coprocessors or specialized devices can communicate with the **CPU** and program running on it by interfacing with this unique memory bus.

1.2.4 Von Neumann Bottleneck

Due to the importance of the system bus in Von Neumann-Derived architectures, it occupies a central place in processor designs, constraining system performance with its physical footprint, transfer speed and energy usage. As circuit speed increase, the amount of data that needs to be transferred not to starve off the processor rises, further straining memory and system buses.

Modern processors make use of multiple levels of hierarchical caches to reduce external memory accesses in order to reduce latency and increase availability of the memory

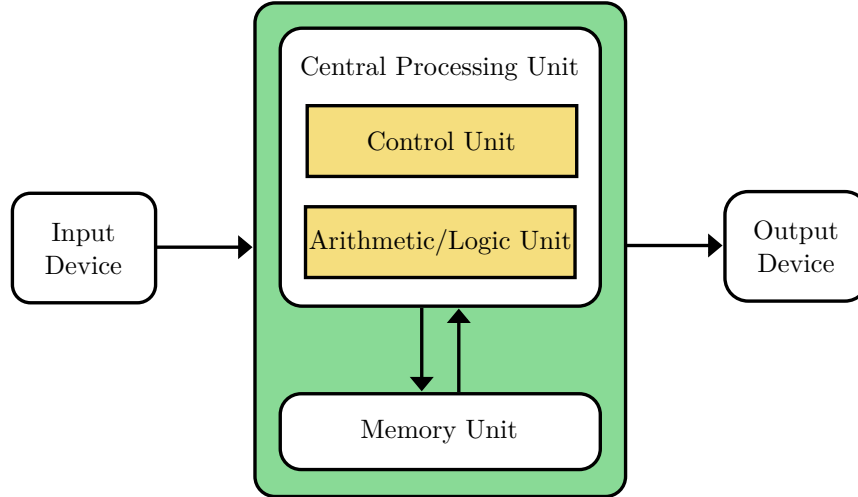


FIGURE 1.1: Von Neumann Architecture diagram, adapted CC-BY-SA work from [Kap13].

buses, while reducing power consumption. However, caching is insufficient, as its effectiveness is highly dependent on the algorithm. Moreover, **Static Random Access Memory (SRAM)** density increase seems to slow down [Sch22] despite cache size continuously growing, attaining multiple gigabytes on commercial offerings through technologies such as 3D stacking [Wuu+22]. The number memory controllers and system buses is also increasing, which makes signal routing more complex, and requires a higher amount of I/O pads [Bec+18].

Another major concern is the increasing fraction of the total available power budget being used to transmit data. Power requirements, measured in J bit^{-1} , increases linearly with bandwidth. This makes data transmission one of the largest sources of power dissipation inside microprocessors [BI13], despite careful management of the energy budget [Bec+18].

Multiple other ways to alleviate this bottleneck are being explored, including non-electrical data transmission using radiofrequency [Cha+08] or optical interconnects [Liu+14], as well as alternative architectures merging processing and memory to reduce the need for data transmission. This thesis focuses on the latter approach.

1.2.5 Ferroelectric HfZrO_2

Hafnium and Zirconium oxide have been employed for years [WSW00] as high permittivity (**high-k**) dielectrics in semiconductor applications, especially below 45 nm, as deployed by Intel in 2007 with its “Penryn” processor line using Hafnium-based **high-k** gate oxides.

In 2006, during the development of dielectric materials for **Dynamic Random Access Memory (DRAM)** capacitor applications at Infineon/Qimonda (Dresden/Germany), a variety of different doped HfO_2 and ZrO_2 thin films were analyzed, revealing nonstandard switching behavior in silicon-doped HfO_2 at some concentrations [SHF19]. Upon further analysis, it was revealed that under some doping and annealing conditions, the material could crystallize into a non-centrosymmetric structure, compatible with ferroelectricity. This led to the first series of publications in 2011 [Bös+11a; Bös+11b; Mül+11], ushering us into a revolutionary era for scaled down ferroelectric devices.

A variety of dopants may be employed to allow hafnium oxides to crystallize into a ferroelectric phase, although Zirconium is generally favored thanks to its low annealing temperature of 500°C [Bou20, p. 144].

Indeed, ferroelectric Hafnium presents multiple advantageous characteristics compared to previous ferroelectric materials:

- **CMOS** compatibility: this is the most interesting property, as hafnium oxide was already widely in use within industrial **CMOS** processes as a gate oxide material.

- Low coercive electric field: from 0.5 MV cm^{-1} to 2.5 MV cm^{-1} [Bou20, p. 145], typically at 1.2 MV cm^{-1} , this allows coercive voltages around 1.2 V for a typical 10 nm oxide thickness, which are compatible with CMOS logic levels.
- Low annealing temperature: at 500°C , HfZrO_2 can be deposited above CMOS devices without damaging them.
- Temperature-stable: Curie temperature can easily surpass 200°C [Bou20, p. 143], depending on doping concentration and transistor size. This is interesting for Non-Volatile Memory applications.

1.2.6 Conclusion

Since the end of Dennard’s scaling circa 2005, power density of integrated circuits has grown, leading to a search for more energy-efficient circuit designs and architectures. This is compounded by the end of Moore’s “law”, where conventional “planar” transistor architectures have reached physical scaling limits. New avenues are being explored for increasing density, with excursions in the 3rd dimension for circuit and transistor designs. However, these approaches also increase the power density, leading to power dissipation issues. At the same time, increasing performance continues to highlight bottlenecks of the Von Neumann architectures and derivatives.

New applications with extremely high performance or power efficiency requirements have also surfaced: genetics, big data applications including machine learning, as well as IoT, smart and embedded sensor networks, etc. These are increasingly using application-specific architectures and devices instead of general-purpose CPUs, now undoing a trend that started in the 1980s. High-performance needs are now served by Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs), as well as other purpose-built ASIC. High energy-efficiency needs are also served by ASICs and FPGAs, with FDSOI transistors.

As a result, interest has been renewed in alternatives to both Von Neumann architectures and CMOS-based circuits.

1.3 3εFERRO European project

The work presented in this document is part of a wider European project: 3εFERRO (standing for *Energy Efficient Embedded Non-volatile Memory Logic based on Ferroelectric Hf(Zr)O₂*). As part its dissemination activities, the project hosts a website at <https://www.3eferro.eu>, and commissioned an introductory video available at <https://youtu.be/M8tL-nN7G-A>, which is effective at conveying an entry-level overview of the goals and context.

1.3.1 Project partners

The project counts eight participants, with various areas of expertise. As such, our team had more opportunity to interact with some partners than others. Our closest collaborators figure at the top of the following list of partners:

1. INL, as part of ECL, Lyon, France
2. NaMLab, Dresden, Germany
3. STMicroelectronics, Grenoble, France
4. CEA-LETI, Grenoble, France
5. EPFL, Lausanne, Switzerland
6. Demokritos, Athens, Greece
7. NIMP, Bucharest, Romania
8. FZJ, Jülich, Germany

1.3.2 Project Goals

This project has a wide scope, reflected in the aforementioned partners. Its main objective is to develop technologies related to the integration of ferroelectric HfZrO_2 with more traditional microelectronics processes.

The approach can be seen as a bottom-up one, starting with material science and characterization of deposited oxides, as well as the optimization of manufacturing process and material properties; then with device and circuit design, characterization and modeling, up to technological demonstrators and architecture performance predictions.

Achievements

This project advanced the state of the art for ferroelectric HfZrO_2 integration.

Deposition methods² were improved[Fra+19b] to optimize the performance of the ferroelectric layer, including the optimization of annealing cycles, temperature and pressure[Bou+19], material and substrate composition[Zac+22]. A lot of effort was spent to improve reliability and yield, to obtain uniform characteristics across devices and wafers, with advanced characterization methods including electrical and microscopic characterization, piezoresponse force microscopy, as well as hard and soft X-ray photoelectron spectroscopy, leading to better models. Device properties were investigated, including **Ferroelectric Capacitors (FeCaps)**, **Ferroelectric Field-Effect Transistors (FeFETs)** and negative capacitance[Gas+19] for use in **Negative-Capacitance Field-Effect Transistors (NCFETs)**. Several circuits and demonstrators were also realized, including 16 kbit memory arrays[Fra+19a; Fra+21].

The project demonstrated the viability of doped Hafnia-based memory devices, and its competitiveness with flash memories in terms of speed, endurance, retention, energy consumption, density, and ease of integration[Gre+20; Oku+21; Alc+22].

Contributions

Our contributions, as outlined in this document, mainly focus on the circuit level and above. They include circuit designs, as detailed in **chapter 3**, as well as the validation of a more complex image filter demonstrator, which is discussed in **chapter 4**. At the highest level, a system-level benchmarking platform was realized to help predict system performance based on the available device-level simulation and characterization data, as described in **chapter 5**.

²Including Atomic Layer Deposition (ALD), Pulsed Laser Deposition (PLD), Physical Vapor Deposition (PVD), Molecular Beam Deposition (MBD)

Chapter 2

Ferroelectrics: behavior and modeling

Contents

2.1 Ferroelectricity	23
2.1.1 Ferroelectric crystals	23
2.1.2 P - V curve	27
2.1.3 Relationship with capacitance and paraelectricity	27
2.1.4 PUND measurements	30
2.2 Modeling	34
2.2.1 Landau model	34
2.2.2 Preisach model	37
2.2.3 Simplified model for large-scale simulation	43
2.3 Ferroelectric capacitors	44
2.3.1 Regular capacitor	44
2.3.2 Non-volatility	44
2.3.3 Negative capacitance	45
2.4 Ferroelectric transistors	45
2.4.1 FeFET Devices	45
2.4.2 Gate stacks	46
2.4.3 Modeling	47
2.5 State-of-the-art on ferroelectric circuits	47
2.5.1 Ferroelectric Hafnia	47
2.5.2 Ferroelectric Capacitors and Back-End of Line circuit design	48
2.5.3 Ferroelectric Field-Effect Transistors-based circuit design	48
2.5.4 Comparison with other Non-Volatile Memories	49
2.5.5 Design-Space Exploration	49
2.5.6 System-level performance evaluation	51

2.1 Ferroelectricity

Ferroelectricity is to electric fields what ferromagnetism is to magnetic fields: when subjected to a strong enough electric field ($E > E_C$), the internal structure of a ferroelectric material rearranges, making it an electric dipole oriented along the external electric field. This polarization is retained even after the disappearance of the electric field, and can later be reversed by applying an electric field in the opposite direction.

2.1.1 Ferroelectric crystals

In response to external stress, the geometry or internal organization of a crystalline lattice is changed, thereby changing the charge density ρ at the surface of the material, if it is composed of electrical dipoles. This change in charge density can in turn affect the electrostatic potential

when measured at the surface of the crystal, depending on the type of stimulus and the material's response.

More precisely, depending on the kind of stimuli that affects a crystal's surface potential or surface charge density, it can be sorted into three nested categories, as illustrated in [Table 2.1](#):

1. piezoelectric crystals change their surface potential in response to mechanical stress;
2. pyroelectric crystals are piezoelectric, and temperature changes also induce a measurable change in their surface potential;
3. ferroelectric crystals are pyroelectric; additionally, the change in surface potential is persistent after applying a strong electric field. The major difference as compared to pyroelectrics is the ability to reverse their polarization before undergoing breakdown [[Ih19](#), p. 7].

The change in surface charge density can be measured by forming a capacitor-like structure with the material placed between two electrodes: a change of voltage should be observed in response to the external stimulus. Intuitively, this can be understood as charges being present in the crystalline structure, and various external stimuli causing the relative charge density to change near the electrodes (thus creating a current), in response to a geometry change. This geometry change can be understood as a response to mechanical stress (directly applied, for piezoelectricity, and indirectly through thermal dilation/contraction for pyroelectrics). Additionally, in the case of a ferroelectric crystal, these charges can reach multiple possible stable locations in the crystal lattice, which gives rise to a bistable behavior.

It should be noted that ferroelectrics retain the new polarization in a stable but reversible way. If the polarization is not retained, they are paraelectrics, as detailed in [subsection 2.1.3](#) and illustrated in [Figure 2.6c](#).

Stimulus	Piezoelectric	Pyroelectric	Ferroelectrics
Mechanical stress	✓	✓	✓
Temperature change		✓	✓
Electric field			non-volatile change

TABLE 2.1: Crystalline classification depending on stimulus causing electrical potential change at the surface. Note that while external electric fields induce a surface charge density change in all of these materials (like a capacitor), ferroelectric materials retain the change after the field dissipates if it exceeds **Coercive Electric field (E_C)**. Moreover, their polarity can be reversed by changing the orientation of the electrical field applied.

Coercive field

Changing a ferroelectric material's electrical polarization requires applying a strong enough electric field, crossing the **Coercive Electric field (E_C)** threshold, and associated with a **Coercive Voltage (V_C)**. This value depends on the ferroelectric material's properties, and correspond to the force needed to move internal charges from one equilibrium position to the other, overcoming internal cohesive forces that arise from a variety of mechanisms, from electrostatic and mechanical effects to chemical bonds. This is illustrated in [Figure 2.2](#) with a spring-charge equivalent system.

The coercive field value is an intrinsic property of the material, though a variety of factors can affect it. An important variation is the domain orientation: as most ferroelectric materials studied are polycrystals, they are made out of multiple crystalline domains with different orientations. As illustrated in [Figure 2.1](#), an external electric field will have less influence on non-aligned crystalline domains, inflating their E_C value. This results in a range of E_C distributed over each domain instead of a single E_C value per ferroelectric device.

Polycrystals and domains

Depending on the material and the crystal growth method, multiple crystalline domains can form. This is the case with current HfZrO_2 deposition methods, especially above a certain size.

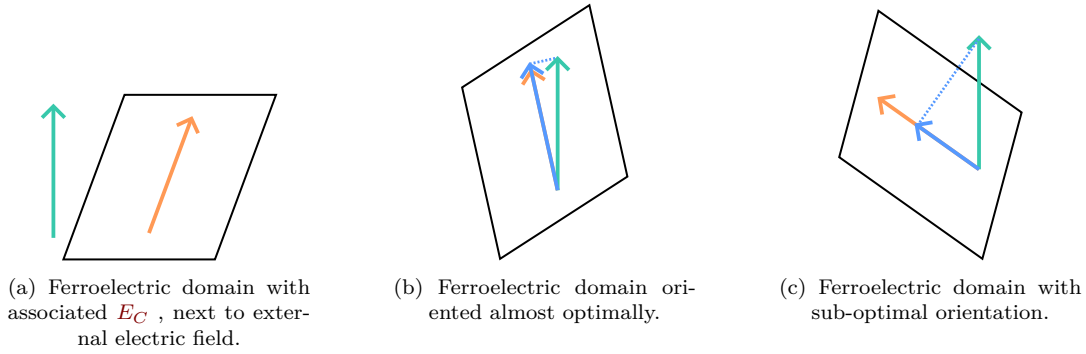


FIGURE 2.1: Effect of orientation mismatch between external electric field and the ferroelectric crystal. 2.1a, 2.1b and 2.1c show the same ferroelectric domain and external field, with different relative orientations. Only 2.1b, which has a Coercive Electric field almost aligned with the external field, can be re-polarized by it. 2.1c would require an electrical field almost twice as strong.

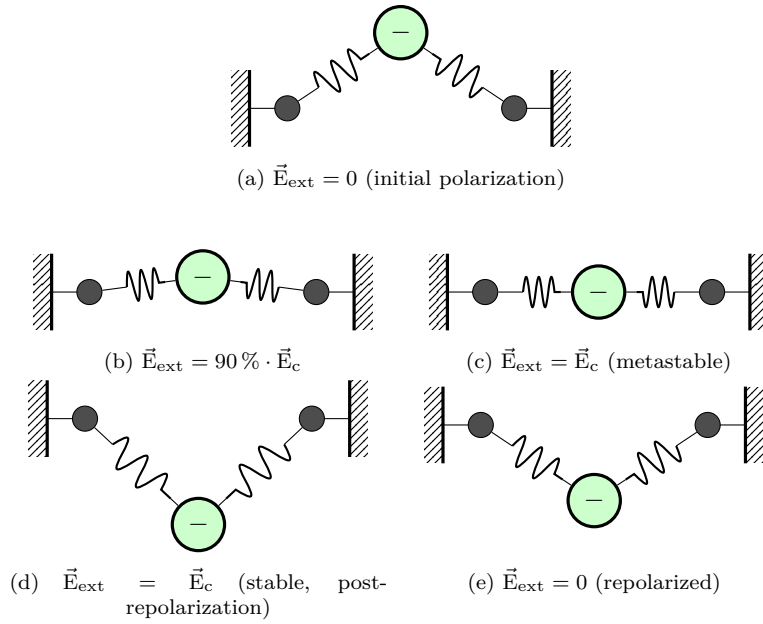


FIGURE 2.2: Illustration of ferroelectricity as a bistable charge-spring system. When compressed or overextended, the springs pull back the charge to one of the two possible equilibrium positions (Figure 2.2a or 2.2e). A strong enough electric field $\vec{E}_{\text{ext}} \geq \vec{E}_c$ (progressively applied from Figure 2.2b through Figure 2.2d) allows to overcome the repulsive force and “flip” the charge to the other equilibrium position, repolarizing the ferroelectric material. This also illustrates how surface charges of a ferroelectric material move to the opposite side during repolarization.

In practice, growing single-domain crystals poses a set of limitations, at least with current HfZrO_2 technology:

- The larger the ferroelectric crystal, the more likely it will split into multiple domains. Therefore, monocrystals must be relatively small, on the order of 10 nm of diameter.
- It is difficult to control the orientation of ferroelectric crystals; therefore, multiple devices will likely have different characteristics, including non-functional ones.
- Even if the aim is to only produce small-size monocrystals, it is extremely likely to have multiple domains in a few devices, especially as the number of devices grows. This variability is detrimental to the yield.

Aiming to only produce monocrystals is therefore currently impractical, mostly due to the difficulty of making the process repeatable across multiple devices: depositing monocrystals of the same size and same orientation across a large number (hundreds to billions) of electrical devices in order to maintain similar electrical characteristics across them is currently unfeasible.

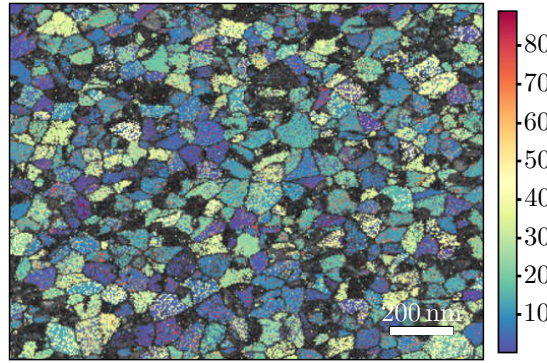


FIGURE 2.3: Experimental domain orientation measurement in polycrystalline Si-doped HfO_2 . Orientation is given in degrees, relative to the image plane. Image CC-BY [Led+20], that quotes an average equivalent diameter of 28.5 nm and 33.9 nm for Zr- and Si-doped HfO_2 , respectively.

Instead, it is preferable to deposit on each device a sufficiently large population of ferroelectric domains so that the behavior is controlled by distribution statistics, which can be made similar across devices. Figure 2.3 illustrates the issue: for the distribution of domain orientations to be similar across random samples, they must be large enough to contain a variety of domains. Using larger polycrystalline devices implies the following:

- To ensure the distribution is the same across devices, they need to be above a certain size to include multiple ferroelectric domains, currently on the order of 200 nm of diameter for HfZrO_2 [Led+20].
- The coercive field value becomes a distribution of multiple values, as it depends on the relative orientation of each domain and the external electric field, as illustrated in Figure 2.1. Such a simulated distribution is shown in Figure 2.14. The effect of increasing the sample size is similar to the increasingly fine-grained distribution shown in Figure 2.16.
- Having different E_C , ferroelectric domains will switch at the different voltages. The polarization will therefore be progressive, making it possible to partially polarize the population, as evidenced by minor loops in the P - V hysteresis of Figure 2.15.
- Minor loops and partial polarization enable the design Multi-Level Cell (MLC)-based memories.
- As domains are more or less receptive to external fields, there can be additional wake-up and fatigue effects.

Note on the nomenclature used for write and erase operations

The terms write (program) and erase (deprogram) are sometimes found in the literature, to designate the direction of an applied coercive field, and therefore the direction of the resulting ferroelectric polarization.

This document will instead explicitly designate the resulting polarization, leaving the meaning of those terms as:

1. Write or program: an operation applying a **Coercive Electric field** over the ferroelectric material. The direction of the field is unspecified, unless it is accompanied by a value being written.
2. Erase: erase operations set the polarization to a known direction, either unspecified, or the one considered the “default” for the memory array. The main distinction with write operation is the purpose, which is to discard the information previously stored.
3. Repolarization: occurs when the ferroelectric material changes its direction of polarization. It can occur after a write or erase operation, but not necessarily so: no repolarization occurs if the polarization was already aligned with the **Coercive Electric field** being applied.

2.1.2 ***P-V*** curve

One of the most widely used tools for characterizing ferroelectrics is the $P = f(V)$ curve, hereafter simply called (***P-V***). This curve represents the electrical charge density at the surface of the material (surface potential P , usually expressed in $\mu\text{C cm}^{-2}$) as a function of the input voltage. Experimentally, it is often obtained by integrating the current measured while subjecting the material to a known voltage, as in the **Positive-Up, Negative-Down (PUND)** measurements from **subsection 2.1.4**. It can also be obtained through other means, such as **Atomic Force Microscopy (AFM)**[Hon+01].

Assuming no leakage currents, integrating the power supply current yields the quantity of charge stored on a capacitor plate, and hence the dielectric’s surface polarization. If the capacitor dielectric is made of a ferroelectric material, an imbalance can be observed during the capacitor’s discharge, if the polarization was changed: as the surface potential rises on one side and falls on the other, electrons will be respectively freed and trapped on the plates, leading to a current peak similar to a temporary leakage current during repolarization. Measuring the number of relaxed charges allows inference of the **remanent polarization (P_r)**, i.e., the quantity of charge trapped at the surface, an important value for comparing ferroelectric materials.

As detailed in **Figure 2.4**, multiple values can be extracted from the ***P-V*** curve:

- **Coercive Voltages**, and by extension E_C .
- **Remanent polarization**, usually read as $2 \cdot P_r$ since this is the height of the curve
- capacitive, resistive and antiferroelectric contributions, to some extent, as shown in **Figure 2.5**, and extracted using **PUND** waveforms as described later in **subsection 2.1.4**.

Limitations of ***P-V*** curves include the fact that they are generated for a given voltage, and that multiple voltage sweeps can yield different **P_r** , manifest as concentric loops. However, this issue is of limited concern as the voltage used is readable on the graph. Conversely, this can be leveraged to show minor loops, where a lower voltage is used deliberately, to only polarize a fraction of the ferroelectric domains, usually for **MLC** operation, or analog control of a **FeFET**’s threshold voltage. Such a minor loop is shown in **Figure 2.15**.

2.1.3 Relationship with capacitance and paraelectricity

Ferroelectric oxides can be considered as dielectrics whose electrical permittivity ϵ_r changes depends on both the current electric field, and the history of the electric field.

A ferroelectric will respond to an external electrical field by changing its internal electrical polarization. They are closely related to paraelectrics and antiferroelectrics, both illustrated

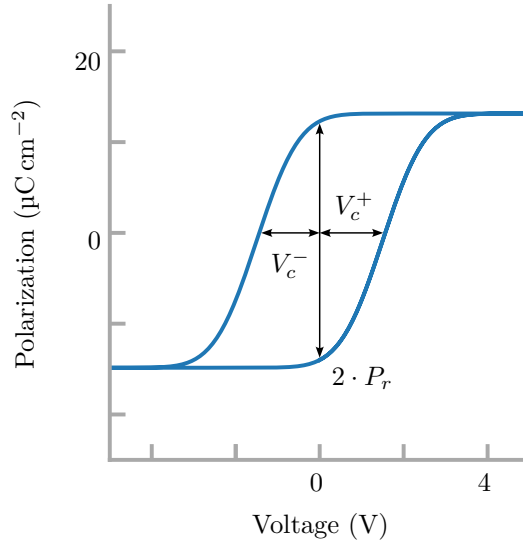


FIGURE 2.4: Reading P_r and V_C on a P - V curve. Note that the V_C and P_r readout will depend on the applied voltage if multiple domains with different characteristics are present. The cycle pictured here is the same as in Figure 2.9b for illustration purposes. The cycle is not necessarily zero-centered horizontally due to **imprint**.

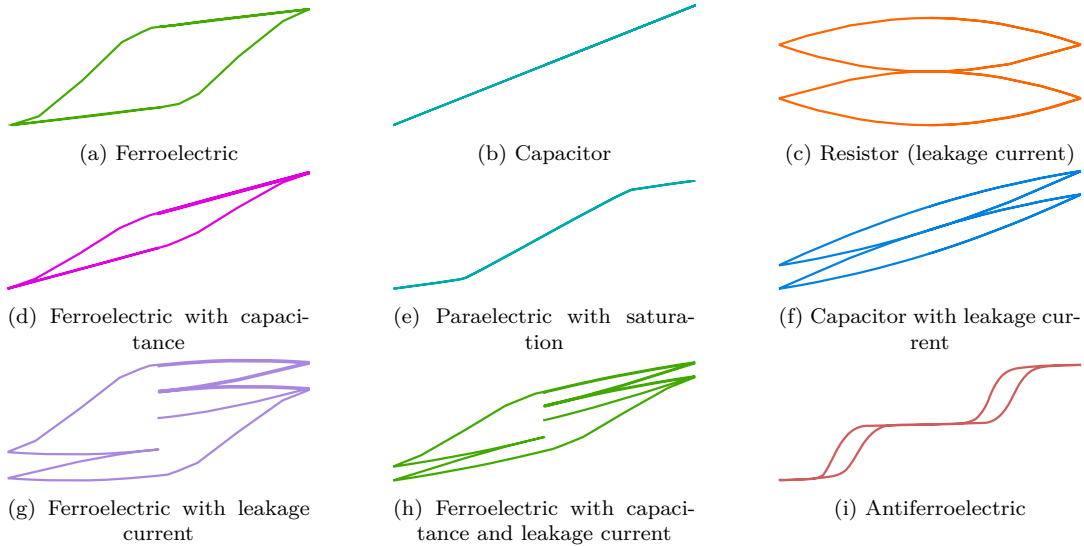


FIGURE 2.5: Sampling of obtainable P - V curves, centered around 0 V. Most of these curves were generated in simulation with $C = 20$ fF for the capacitance, $R = 100$ G Ω for the leakage current, under a 50 Ω power supply. It should be noted that the timescale used for simulation was much greater than the various τ , hence approximating steady-state.

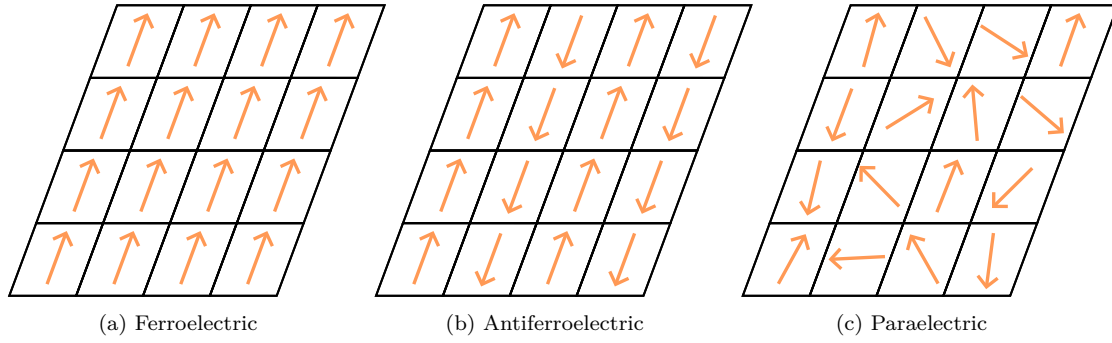


FIGURE 2.6: Electrical polarization orientation of domains after application of a strong (coercive) external field. The figure represents multiple crystalline domains of three different materials. Note that the ferroelectric material (2.6a) is aligned with the external field, the paraelectric (2.6c) material went back to its random initial orientations, and the antiferroelectric material (2.6b) spontaneously compensated the new orientation.

in Figure 2.6 and Figure 2.5. However, in contrast with those, they retain a non-zero global polarization when no external electric field is present.

As dielectrics, all of these material classes can be used as an insulator between capacitor plates. Their internal response to the externally applied electric field will reinforce the charge buildup at the surface of the electrodes, thereby increasing the capacity and providing a faster charging rate. The exact behavior depends on the material class:

- For paraelectrics, the behavior is similar to that of simple capacitors, with internal charge mobility providing a boosting effect to the capacitance, providing a higher effective ϵ_r . Charge mobility may be constrained inside the crystalline structure of the material, which will limit the paraelectric effect, and will lower the capacitance value after a certain field strength, until the dielectric breaks down.
- For ferroelectrics, as long as E_C is not reached or exceeded, the capacitor will function as a normal paraelectric one. However, upon reaching E_C :
 - if the domains were already polarized in the same direction as the applied electric field, there is no repolarization, and the capacitor continues to function as a normal capacitor below and above E_C .
 - if the domains were not already polarized in the same direction as the applied electric field, they will start orienting their polarization along the external field, starting with those with a lower coercive field (i.e., better aligned with the external field). That process releases a large quantity of charges, which corresponds to a much steeper P - V curve, as can be observed on the sides of Figure 2.4. This occurs as long as the voltage continues to increase, and until most domains are polarized.

As with paraelectrics, the charge mobility providing enhanced capacitance can reach its limits, and decrease capacitance until breakdown.

- For antiferroelectrics, the paraelectric effect is naturally compensated by a secondary electrical dipole, which nullifies the overall polarization of the material, resulting in a theoretically null (0 F) capacitance, and consequently zero surface charges. That compensating dipole itself can, however, be overturned under sufficiently strong electric fields, provided the dielectric does not break down. This gives rise to smaller hysteresis cycles on either side of the P - V curve, as shown in Figure 2.5i.

Charge screening and depolarization field

Changing the ferroelectric polarization brings charged particles closer to the surface of one of the electrodes, and away from the second, which changes the charge density P at the surface

of the material. This means that both electrodes have a non-zero (and of opposite value) electric charge after a change of polarization.

As a result:

1. An electric field exists across the **FeCap** after polarization: the depolarization field E_{dep} ;
2. Mobile charges close to the electrodes will be attracted or repulsed to compensate, or “screen” the new charge, as shown in **Figure 2.7**;
3. Charges “captured” for screening the previous polarization are released, which creates a measurable repolarization current.

The resulting depolarization field is, as its name suggests, oriented in the direction opposite to the recently applied field, and as such will attempt to undo the polarization. This can be a significant source of retention loss, particularly when E_{dep} is much larger than E_C . **HfZrO₂**-based ferroelectric devices having a larger coercive field than more traditional **Pb(Zr, Ti)O₃** (**PZT**) or **SrBi₂Ta₂O₉** (**SBT**) ferroelectric material, retention loss due to the depolarization field is much lower[**MG19**], though E_{dep} is larger in thinner films[**Maj22**].

The second mechanism prevents measuring ferroelectricity on a macroscopic scale. It has been suggested[**CL07**] that this charge screening effect is responsible for the much later discovery of ferroelectricity compared to ferromagnetism. Indeed, the screening distance is relatively small, and varies from 0.1 nm in metal to a few nm in less conductive materials, according to [**PLH21**].

This charge screening behavior has very localized effects, and as such plays a role in the **FeFETs** structures described later in **section 2.4**. Charges screening the polarization are not mobile, and thus may not change the conductivity or **threshold voltage** (V_{th}) of **MOSFETs** if they are located in the channel[**PLH21**]. Electrons temporarily trapped in the oxide can also screen charges, degrading the performance of **FeFETs**, thus creating read-after-write issues where the new **FeFET** polarization cannot be read immediately after writing it[**Kle+21**]. This trapping has significantly higher chances of occurring when hot carriers (due to higher voltages used during programming) tunnel first across the ferroelectric, and are blocked at the interfacial layer by the depolarization field. A de-trapping pulse can be applied if fast read-after-write is necessary[**Mul+21**].

The last important effect is the release of charges screening the previous, opposite polarization: this repolarization current allows **PUND** measurements as introduced in **subsection 2.1.4**, and enables multiple destructive reading mechanisms generally used in **FeCap**-based designs detailed in **chapter 3**.

Ferroelectric Tunnel Junction

While this effect is not the main focus of this document, a leakage current exists through the ferroelectric layer, especially at low thickness, due to the tunnel effect[**GB14**]. The ferroelectric polarization changes the density of charges at the surface, which are compensated (“screened”) by displacing charges from the electrode material. This compensation can be very localized, or occur over longer distances, which significantly reduces the tunneling current as normally conducting materials are locally deprived of charge carriers. Therefore, a design such as the one illustrated in **Figure 2.7e** and **Figure 2.7f** with asymmetrical electrodes (and thus different charge screening lengths) can significantly modulate tunneling current with the ferroelectric polarization, allowing the polarization state to be retrieved without repolarizing the ferroelectric. This is leveraged in the **Ferroelectric Tunnel Junction (FTJ)** operation mode of the 2T1C circuit presented in **section 3.5**. Electrodes consisting of metal and semiconductor seem to yield the strongest current-polarization dependency[**GB14**; **Maj+18**; **Maj22**, pp. 10].

2.1.4 PUND measurements

To answer the need for independent measurement of the electrical characteristics of the paraelectric and ferroelectric parts of a ferroelectric capacitor, a **PUND** waveform can be used[**Mül+11**; **SG96**]. As the acronym suggests, this measurement strategy is split into four parts. A specific voltage waveform is applied, while current is monitored:

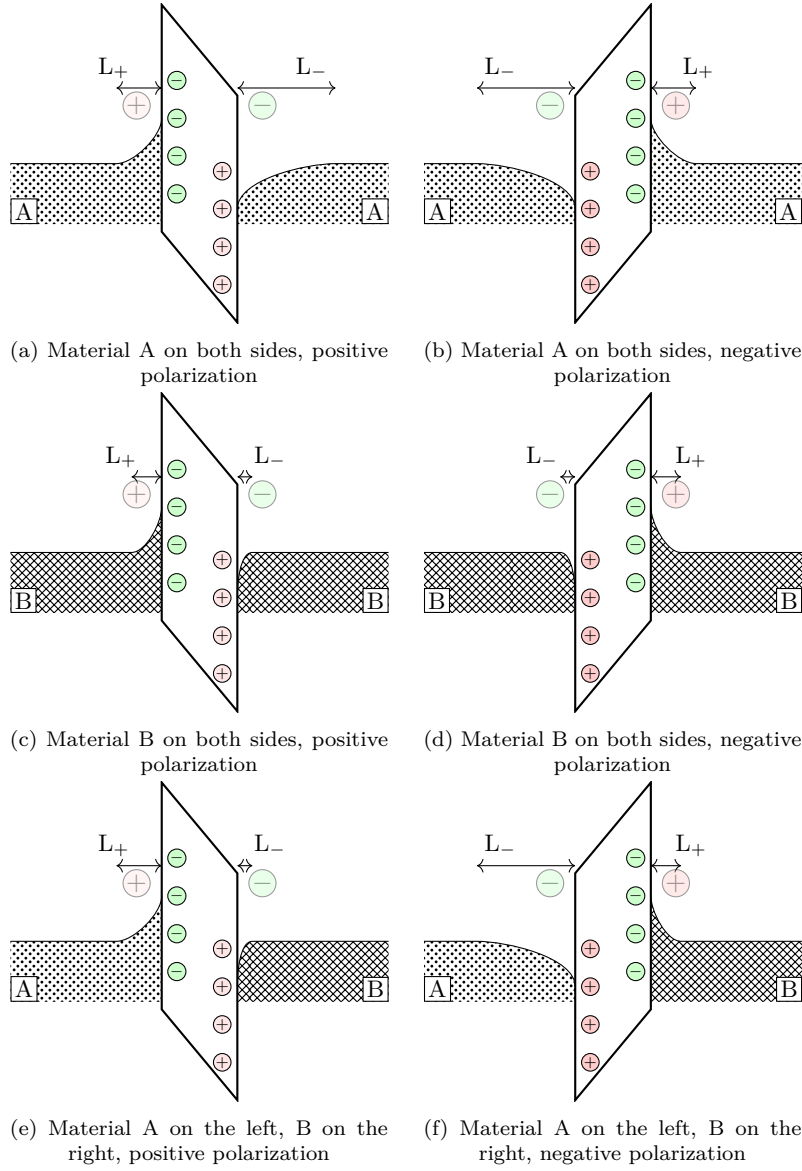


FIGURE 2.7: Band diagrams with screening lengths L_+ and L_- on both sides of the ferroelectric layer, corresponding to two materials A and B. The diagram is displayed at rest (0 V) after polarizing with $+V_C$ and $-V_C$ in the left and right column, respectively (voltage reference taken on the right side). Notice how $L_+ + L_-$ is constant regardless of the polarization when both sides are made of the same material, and becomes polarization-dependent in an asymmetrical assembly.

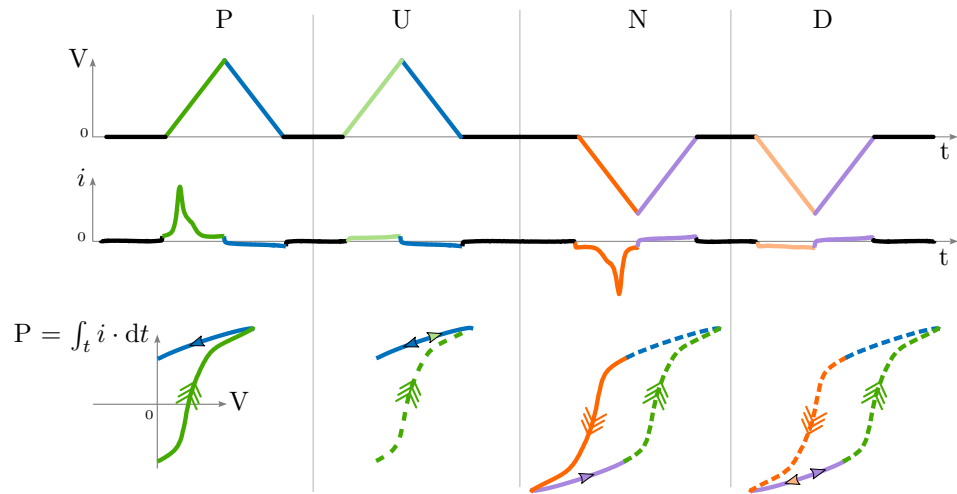


FIGURE 2.8: **PUND** P - V waveform (top), with associated $I = f(t)$ current (middle), and the associated uncorrected $P = f(V)$ hysteresis cycle being constructed. Colors have been used to represent:

- the polarization phase with a positive (**P**) voltage
- the secondary “up” (**U**) phase, with only the leakage and capacitor charging currents
- the second polarization phase with a negative (**N**) voltage ramp
- the last phase where voltage is brought down (**D**) again to measure leakage and capacitor charging currents in that direction
- as well as the “return to 0V” phases (respectively from above and below), that are the same across P and U, as well as N and D regardless of the number of phases, as the polarization never changes during these.

1. A positive voltage ramp switches the polarity of the ferroelectric capacitor to the “positive” state (sometimes referred to as “programming” in the literature, as discussed in [section 2.1.1](#)). The slew rate is slow enough for the ferroelectric polarization to be completely switched. Voltage ramps back down to zero.
2. The same ramp is applied again. This time, however, no ferroelectric switching occurs as it already occurred in the previous step – i.e. domains whose polarization would change under such an electric field have already done so during the first ramp. This step allows the measurement of the regular, paraelectric capacitance, as well as leakage currents. Only paraelectric domains align with the “up” polarization of the electric field.
3. The process is repeated to measure the negative part of the switching diagram: the same voltage ramp is applied, with its polarity reversed, to reverse the polarization of the previously switched ferroelectric domains. A “negative” polarization is written to the ferroelectric oxide during this phase.
4. Once again, the same negative-polarity ramp as in the previous step is applied to measure the paraelectric and leakage contribution to the measurements. Paraelectric domains align with the “down” direction of the electric field, and return to their rest position once the externally applied voltage returns to zero.

The resulting waveform is plotted in [Figure 2.8](#). The above sequence is usually repeated twice, or at least preceded by a negative pulse, in order to set every ferroelectric domain to a known state before performing a measurement cycle. This in turn renders the extracted cycle continuous (and enables perfect looping of the P - V curve).

The current is measured during the voltage sequence described above. The resulting current data contains:

1. the switching current during steps 1 and 3,
2. the leakage current, which is present during the whole cycle,
3. the capacitor charge/discharge current, also present during the whole cycle.

The switching current conveys charges stored in the ferroelectric capacitor, so the surface polarization can be found by integrating this quantity: $P(t = T) = 1/A_{CFE} \cdot \int_{t=0}^T i(t) \cdot dt$. It is usually expressed in $\mu\text{C cm}^{-2}$.

If the above procedure is carried out with the current measured during the first pulse of each polarity, the resulting P - V ferroelectric response will be skewed by the paraelectric capacitor contribution (with a linear P - V relation), as well as the leakage current. The second pulses in turn contain only the leakage and paraelectric response, as the ferroelectric response can only be triggered again by depolarizing the ferroelectric domains, which is done in the second half of the PUND cycle when polarity is reversed.

By subtracting the second set from the first, the pure ferroelectric, unskewed cycle can be observed, as plotted in [Figure 2.9](#). This can be observed through the contrast of [Figure 2.9a](#) with [Figure 2.9b](#). The first graph is obtained with:

$$P(t) = \int_{t=0}^{t=N} i(t) \cdot dt$$

assuming Δt separates both the P and U, as well as the N and D cycles, [Figure 2.9b](#) is generated from:

$$P(t) = \int_{t=0}^{t=\Delta t} i(t) - i(t + \Delta t) \cdot dt \quad (t \notin U, D)$$

or in simpler terms, with i_X the current measured during the phase X :

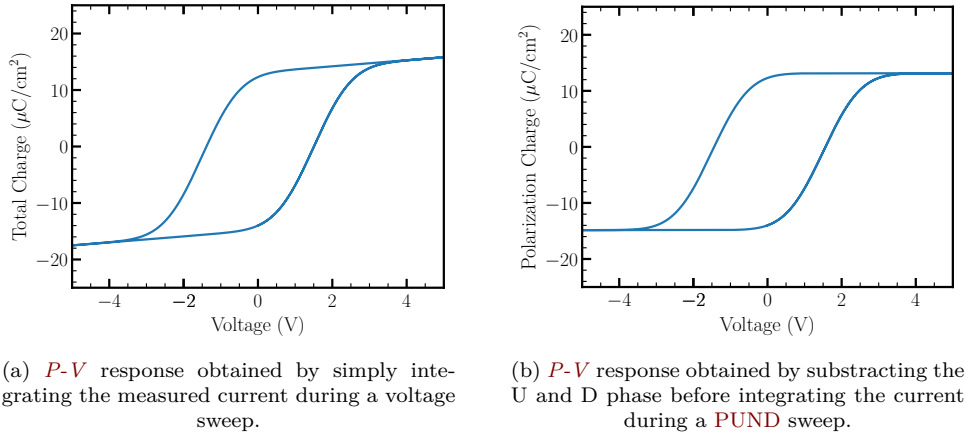


FIGURE 2.9: P - V response skewed by capacitance (2.9a), and corresponding pure ferroelectric response (2.9b) extracted through PUND stimulation. Simulation from the idealized ferroelectric distribution shown in Figure 2.14, where V_C starts at 0 V.

$$\begin{cases} P(V) = \int_V i_P(V) - i_U(V) \cdot dV, & \text{PU phase} \\ P(V) = \int_V i_N(V) - i_D(V) \cdot dV, & \text{ND phase} \end{cases}$$

2.2 Modeling

All simulations were performed with Cadence® Spectre®.

There are various approaches to modeling, depending on the required accuracy. More accurate models are useful for validating small circuits such as logic gates, and ensure that ferroelectric materials can be programmed reliably. A simplified model can then be used on a larger scale to validate a full circuit, as will be described in subsection 4.6.4.

In this section, we will detail three modeling approaches:

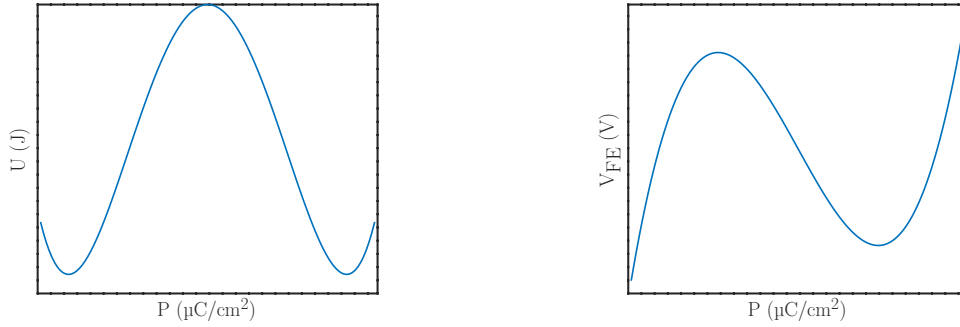
1. In subsection 2.2.1, the idealized Landau-Kalathnikov phase transition model provides some insight into ferroelectric switching mechanisms. It is usable only on single ferroelectric domains, and must therefore be only used on small ferroelectric device sizes.
2. In subsection 2.2.2, the Preisach modeling approach, which is more applicable to a statistical distribution of ferroelectric domains, is presented. As such, it should be used when considering larger ferroelectric domains, preferably those including a large and homogeneous population.
3. Finally, in subsection 2.2.3, a few approaches to simplifying models for large-scale simulations will be given.

2.2.1 Landau model

Description

The Landau model is well-suited for explaining the switching behavior of individual ferroelectric crystals. It is a symmetry-based approach of the Gibbs free energy equation, applied to phase transition problems.

First applied to ferroelectrics by Devonshire[CL07; Dev49], the existence of two equilibrium positions visible in Figure 2.10a makes it well-suited for mono-domain simulations. The mathematical equation is obtained using a Taylor expansion of the Gibbs free energy equation, then simplified using symmetry considerations, and assuming some parameters to



(a) Gibbs free energy equation when applied to the phase transition problem of a switching ferroelectric material. Note the two potential wells that correspond to the equilibrium positions of both possible ferroelectric orientations.

(b) Derivative of plot from Figure 2.10a. This plot is more directly relevant to circuit design, as it exhibits the electrical field or electrical potential across the ferroelectric material.

FIGURE 2.10: Simulated Gibbs free energy equation (2.10a) and derivative (2.10b) for coefficients published in the literature[Yin+16] for HfZrO_2 : $\alpha = -7 \times 10^9 \text{ m F}^{-1}$, $\beta = 3.3 \times 10^{10} \text{ m}^5/\text{F/C}^2$, $\gamma = -0.2 \times 10^{10} \text{ m}^9/\text{F/C}^4$, thickness $t_{FE} = 5.7 \text{ nm}$.

be static (independent of time and input frequency), which limits the number of degrees of freedom[CL07].

Expression

The Gibbs free energy equation $U = f(P)$ gives the energy of the surface as a function of surface polarization. A sixth-order expansion is generally regarded as being sufficient to describe the system, with the symmetry consideration allowing the removal of half the terms[CL07; SD08; WA17]. Along a single dimension, it can be expressed as:

$$U = \alpha P^2 + \beta P^4 + \gamma P^6 - \vec{E}_{ext} \cdot \vec{P}$$

$$\frac{dU}{dP} = \nabla_{\vec{P}} U = 2\alpha P + 4\beta P^3 + 6\gamma P^5 - E_{ext}$$

Applying the phenomenological Landau-Ginzburg-Devonshire theory to describe dynamical properties of ferroelectrics leads to the following Landau-Khalatnikov equation[FKK12; Mas+21]:

$$\rho \frac{d\vec{P}}{dt} + \nabla_{\vec{P}} U = 0 \quad (2.1)$$

By making use of the Landau-Khalatnikov Equation 2.1:

$$0 = 2\alpha P + 4\beta P^3 + 6\gamma P^5 - E_{ext} + \rho \frac{dP}{dt}$$

$$E_{ext} = 2\alpha P + 4\beta P^3 + 6\gamma P^5 + \rho \frac{dP}{dt}$$

As a first approach, we can consider the static case (this also allows us to skip Equation 2.1 if we assume a uniform polarization, in which case $\nabla_{\vec{P}} U = 0$):

$$E_{ext} = 2\alpha P + 4\beta P^3 + 6\gamma P^5 \quad (2.2)$$

t_{FE} is ferroelectric thickness, so $V_{ext} = E_{ext} \cdot t_{FE}$:

$$V_{ext} = 2\alpha t_{FE} P + 4\beta t_{FE} P^3 + 6\gamma t_{FE} P^5$$

With A the capacitor area and Q the number of charges, such that $P = Q/A$:

$$\begin{aligned}
 V_{ext} &= 2\alpha t_{FE} \frac{Q}{A} + 4\beta t_{FE} \frac{Q^3}{A^3} + 6\gamma t_{FE} \frac{Q^5}{A^5} \\
 V_{ext} &= \frac{1}{A} \left(2\alpha t_{FE} Q + 4\beta t_{FE} \frac{Q^3}{A^2} + 6\gamma t_{FE} \frac{Q^5}{A^4} \right) \\
 \frac{1}{C_{FE}} &= \frac{dV_{ext}}{dQ} = \frac{1}{A} \left(2\alpha t_{FE} + 12\beta t_{FE} \frac{Q^2}{A^2} + 30\gamma t_{FE} \frac{Q^4}{A^4} \right) \\
 \frac{1}{C_{FE}} &= \frac{1}{A} (2\alpha t_{FE} + 12\beta t_{FE} P^2 + 30\gamma t_{FE} P^4)
 \end{aligned}$$

This results in an expression of the ferroelectric capacitance:

$$\frac{A C_{FE}}{C_{FE}} = 2\alpha t_{FE} + 12\beta t_{FE} P^2 + 30\gamma t_{FE} P^4 \quad (2.3)$$

Usage

This expression can then easily be used in conjunction with a capacitor model, and can be considered as equivalent to a capacitor in series with a voltage generator, the voltage of which depends on the quantity of charges accumulated in the capacitor. This approach also translates to **FeFET**[WA17], where the ferroelectric layer is used either directly as the gate oxide, or stacked on top of it, in which case it is in series with the gate capacitor, as described in [subsection 2.4.3](#).

In actual devices, the ferroelectric layer is partially ferroelectric, and partially paraelectric (purely capacitive), due to both electrodes, and non-ferroelectric domains, as can be evidenced by **PUND** waveforms, detailed in [subsection 2.1.4](#). This ferroelectric capacitance is therefore more accurately used in parallel with a regular capacitor.

Fitting

Fitting the model to experimental data is relatively easy, as it consists of a simple polynomial fit, with a few coefficients constrained to be zero. It should however be stressed that this model is not designed to accurately represent capacitor behavior, implying therefore that experimental data should be obtained with a **PUND** stimulus to remove the paraelectric and leaky capacitor response from the purely ferroelectric response: although the first term of the polynomial can fit an ideal capacitor with a linear $1/C_{FE} = dV/dQ$, this task is best left to a specialized model, leaving the ferroelectric model to be fitted to experimental data free of parasitics from a real, non-linear capacitor. A sample fit on experimental data obtained at **EPFL** is displayed in [Figure 2.11](#). Corresponding **GNU Octave** (**MATLAB**®-compatible) code for performing that fit is provided in [Listing A.7](#). The line numbers provided in the next paragraph refer to this code listing.

To automate the fitting process, and since the **S-curve** is not directly observable under characterization, the vertical parts (represented with dots on [Figure 2.12](#)) of the hysteresis cycle must be discarded. This is illustrated in [Figure 2.11a](#), where **regions of interest (ROIs)** are identified. These **ROIs** contain the only points that the polynomial is fitted on, and are selected as follows:

1. Select the two endpoints of the cycle. These are the coordinates corresponding to the highest voltage values ([Line 34](#)).
2. Select the two inflection points of the cycle:
 - (a) Compute the bisector line of the cycle, which is constructed as the line that connects the two previously-defined endpoints ([Line 53](#))
 - (b) Find the points of the cycle that are furthest from this line, by orthogonally projecting them ([Line 72](#)).
 - (c) The furthest point in each half is an inflection point ([Line 85](#)).

3. The two **ROIs** are those that connect the aforementioned points of interest in the same half of the curve, through the shortest path along the voltage axis (**Line 110**).

The polynomial curve is then fitted with the axes reversed as $V = f(P)$, as opposed to the more generally used “**S-curve**” $P = f(V)$ P - V plot. This fit is made while constraining even coefficients to zero in order to satisfy **Equation 2.2**, as the provided code in **Listing A.9** does.

Conclusion

While this model is of limited use, due to being constrained to single-domain crystals, it does allow the negative capacitance aspect of ferroelectric devices to be explored, and is usually leveraged for simulating **NCFETs**[**SR17**]. It is also perfectly usable as a first approach to simulating ferroelectric circuits using polycrystalline ferroelectrics, provided that such circuits do not make use of precise analog control of the electrical polarization[**Azi+18**].

This model also contains a dynamic component that was not studied here, but which could be of interest for dynamic simulations. The two important model equations are numbered **2.2** and **2.3**. They introduce material-dependent *Landau* coefficients α , β and γ ; while describing the material’s polarization (and therefore the charges it releases when the polarization changes) as a function of an externally applied external field or voltage.

2.2.2 Preisach model

The Preisach[**Pre35**] model uses a mathematical analysis of an experimentally-obtained hysteresis cycle, such as that pictured in **Figure 2.11** or in **Figure 2.12**. Such cycles can be estimated using a sum of “hysterons”, which are simple mathematical hysteretical functions.

Originally devised for ferromagnetism, this approach can be applied to ferroelectricity, and allows the modeling of multiple ferroelectric domains, which enables the use of minor loops in the ferroelectric cycle, for instance in **MLC** applications[**KN03**].

However, this model becomes less accurate when fewer ferroelectric domains interact, which is the case with smaller feature sizes. According to **NaMLab**, their implementation should not be used with ferroelectric capacitors of diameters below 150 nm, i.e., the limit of validity of this model is estimated to be around 150 nm, which corresponds to the limit below which individual ferroelectric domains have a measurable effect.

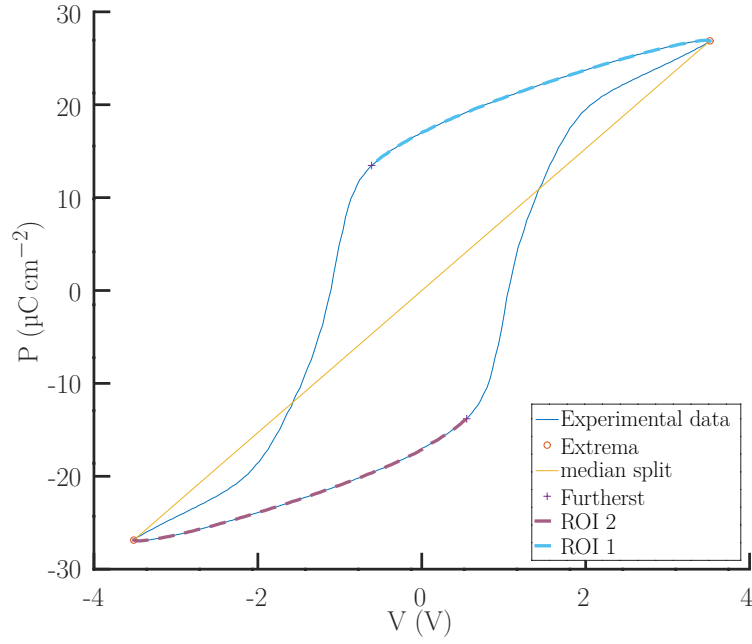
Acknowledgements

This section received major contributions from Damien Deleruyelle, who wrote a Preisach model to generate the figures.

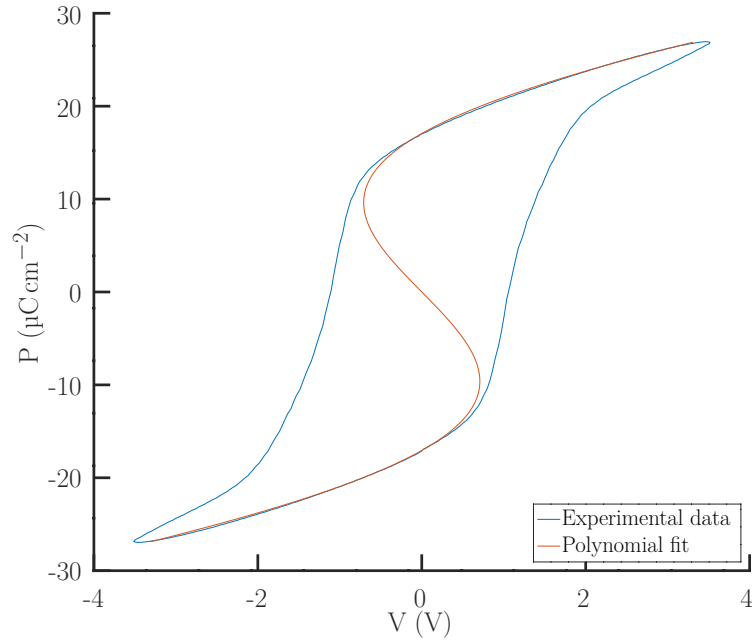
Hysterons

The base principle behind the Preisach model is that the ferroelectric material consists of a set of individual dipoles that contribute to the total polarization. Each dipole has two individual coercive voltages (V_c^+ and V_c^-), which are required to make it switch in the opposite direction. That polarization reversal is assumed sharp and continuous, and cannot be interrupted. Thus, each dipole has a rectangular hysteresis loop (hystreron), as shown in **Figure 2.13**. Based on the assumption that the dipoles do not interact with each other, the hysteresis loop of the macroscopic system is seen as a superposition of these hysteresis units. The coercive voltages of each hysteresis unit in the macroscopic system are supposed to be statistically distributed: this condition limits the use of the Preisach model for smaller devices. The distribution of *up*- and *down*-states among the dipoles (thereafter noted μ) depends on the applied voltage, as well as the voltage history, and the coercive voltages for the considered dipole $\mu_{V_c^+, V_c^-}$. This means that possible “turning points” (reversal of the direction of evolution) of the voltage curve will directly impact this dipole state distribution μ .

The hystreron behavior of each dipole can be modeled mathematically with the following equation:



(a) Identification of ROIs for fitting experimental data



(b) Polynomial fit of degree five constrained to odd powers compared to experimental data

FIGURE 2.11: Selecting ROIs from experimental data (2.11a). Only these regions will be used for fitting the polynomial plotted in 2.11b, whose equation is found to be $V = -0.111695P + 0.000423615P^3 - 1.36673e-07P^5$, or equivalently, extracted Landau coefficients as $\alpha = -3.49047 \times 10^8 \text{ m/F}$, $\beta = 6.61898 \times 10^9 \text{ m}^5/\text{F/C}^2$, $\gamma = -1.42368 \times 10^{10} \text{ m}^9/\text{F/C}^4$.

Note that the fit was not performed on a PUND-extracted P - V curve, due to lack of experimental data. These plots were directly generated from Listing A.7.

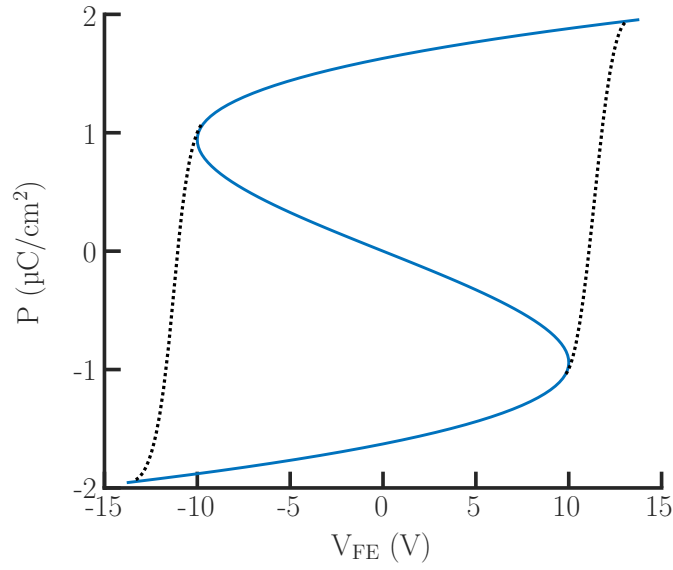


FIGURE 2.12: Example ferroelectric hysteresis cycle, with the same parameters as Figure 2.10. This displays the same graph as that from Figure 2.10b, but with the axes permuted to display the telltale ferroelectric “S-curve”. Also visible in dotted lines is the path taken by the ferroelectric oxide when subjected to a monotonically increasing (resp. decreasing) voltage.

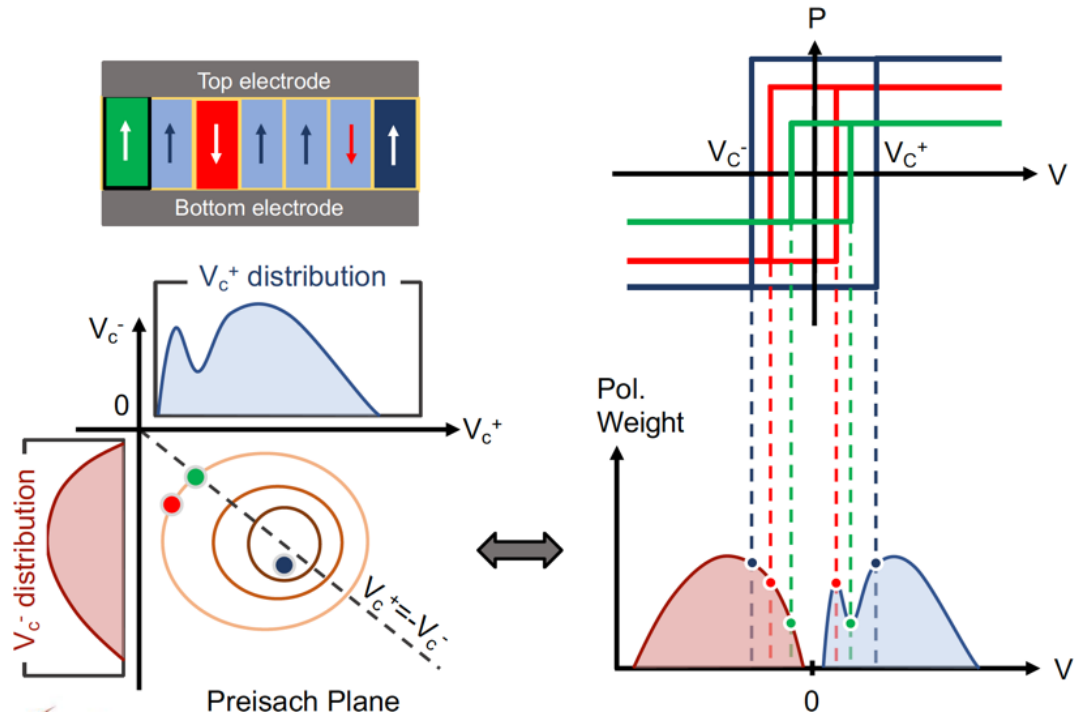


FIGURE 2.13: Preisach model – hysterons and plane

$$\mu_{V_c^+, V_c^-}(V(t)) = \begin{cases} -1, & V(t) \leq V_c^- \\ \gamma, & V_c^- \leq V(t) \leq V_c^+ \\ +1, & V(t) \geq V_c^+ \end{cases}$$

where the value of γ is determined by the previous history of the applied voltage. If the last time $V(t)$ was outside the range $[V_c^+, V_c^-]$ while being greater (resp. lower) than V_c^+ , then $\gamma = 1$ (resp. -1). In this equation, no switching delay is assumed, and each hysteron is weighted through the $(V_c^+$ and $V_c^-)$ distribution. While the above equation assumes a square step-function, this may not be the most physically accurate, nor the most mathematically convenient. Moreover, the discontinuity introduced by the step function might cause instabilities in simulation. For these reasons, the Miller variant[Mil+90] uses a tanh step-function.

Denoting V_m as the magnitude of the maximal voltage dynamics applied to the macroscopic ferroelectric capacitor, and assuming a normalized $(V_c^+$ and $V_c^-)$ distribution, namely ρ , we can write:

$$\int_{-V_m}^{V_m} \int_{-V_m}^{V_m} \rho(V_c^+, V_c^-) \cdot dV_c^+ \cdot dV_c^- = 1$$

Therefore, 2D Gaussian distributions for $(V_c^+$ and $V_c^-)$ as shown in Figure 2.14 are good candidates to describe hysteron distribution.

Cumulative hysteron behavior

The variation in polarization is computed by integrating $\rho \cdot \mu$ over the Preisach plane, that is integrating the state of each hysteron (or rather, every part of the distribution) over the voltage history:

$$P(V(t)) = \iint_D \rho(V_c^+, V_c^-) \cdot \mu_{V_c^+, V_c^-} \cdot V(t) \cdot dV_c^+ \cdot dV_c^-$$

where D depends on the trajectory of $V(t)$ such that:

$$\begin{cases} D^+ = [V_i; V_f] \times]-\infty, +\infty[, & \text{if } \frac{dV}{dt} > 0 \text{ (increasing voltage)} \\ D^- =]-\infty, +\infty[\times [V_i; V_f], & \text{if } \frac{dV}{dt} < 0 \text{ (decreasing voltage)} \end{cases}$$

In these equations, V_i (resp. V_f) is the initial (resp. final) value of $V(t)$. It is also possible to compute internal loops, which are due to changes of direction in the input voltage, within the $[-V_m, +V_m]$ range, and which also leads to voltage turning points, as shown in Figure 2.15.

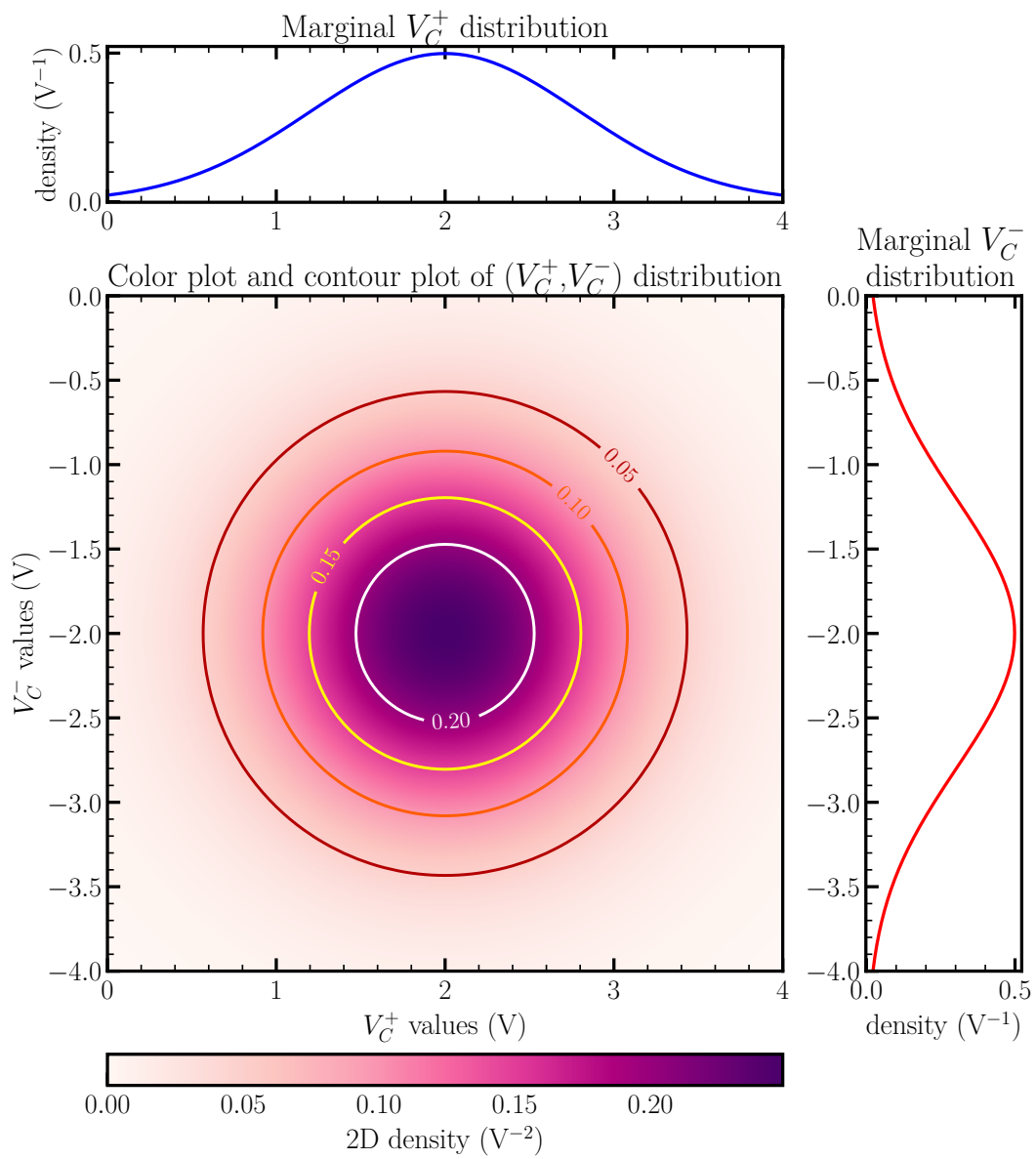
Interestingly, the (V_c^+, V_c^-) distribution ρ can be theoretically extracted from incremental polarization cycles δP (i.e. $\rho \cdot \mu \cdot \delta V_c$), which can be deduced by differentiating consecutive backward P - V curves during so-called **First Order Reversal Curve (FORC)** measurements[Peš+17]. This is illustrated in Figure 2.16, and may be leveraged for experimentally determining the distribution of coercive voltages among ferroelectric domains.

Given the flexibility of this numerical approach, arbitrary (non-Gaussian) $\{V_c^+, V_c^-\}$ distributions can also be employed or extracted, as shown in Figure 2.17.

Limitations

This model is currently one of the most employed for circuit simulation. However, while it is able to represent different variability of ferroelectric domains with multiple distributions, it is unable to model **FTJ**, negative capacitance, as well as accumulative and stochastic switching behaviors[Den+20]. Moreover, this model, at least as a first approach, needs to track the state of the hysteron population, which causes scalability issues, especially with larger circuits containing multiple ferroelectrics devices.

¹Both these integral bounds can be 0 in the typical case where $V_c^- < 0$ and $V_c^+ > 0$.

FIGURE 2.14: (V_c^+, V_c^-) 2D Gaussian distribution

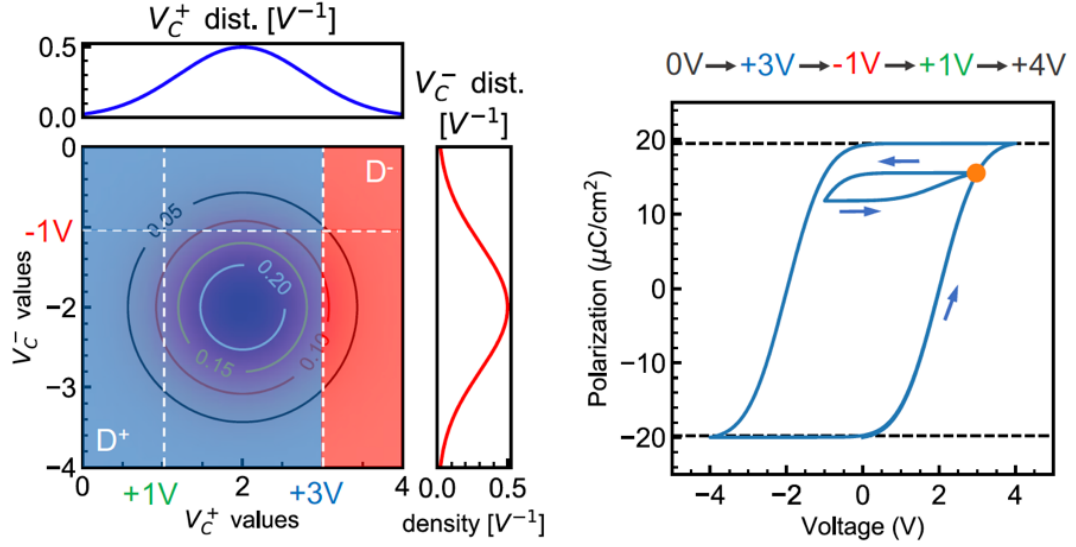
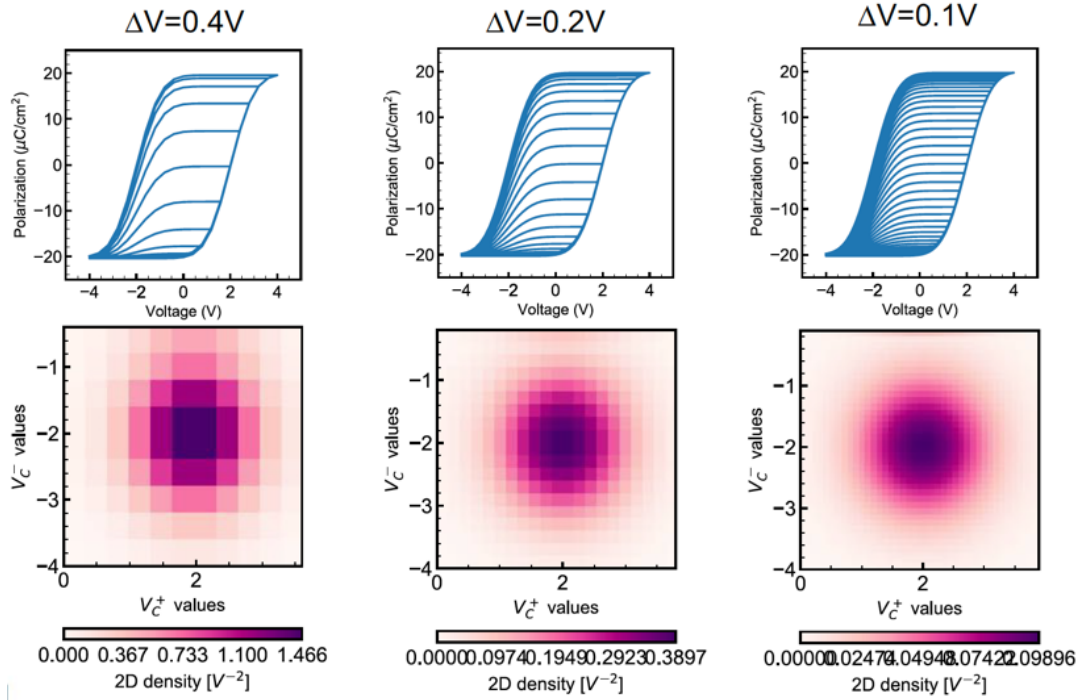
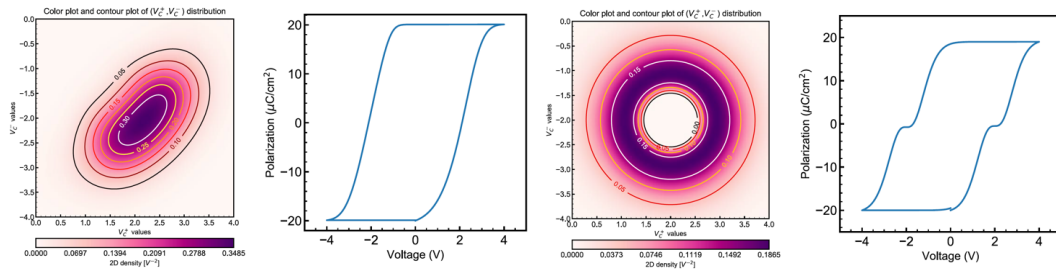


FIGURE 2.15: Internal loops and turning points.

FIGURE 2.16: Example of 2D V_c^+/V_c^- Gaussian distribution extraction with increasingly fine-grained V_c voltage resolution. Increasing the number of crystalline domains has a similar effect on the distribution.FIGURE 2.17: Handling arbitrary V_c^+/V_c^- distributions

LISTING 2.1: Minimal Verilog-A ferroelectric model for use in FeFET simulations as a computationally lighter alternative to a more realistic one.

```

1  `include "constants.vams"
2  `include "disciplines.vams"
3
4  module miniferro(in, out);
5  input in;
6  output out;
7  electrical in, out;
8
9  parameter real vc = 2; // volts
10 parameter real vth = 0.4; // volts
11 parameter real vout = 1.0; //volts
12
13 integer is_on = 1;
14
15 analog begin
16
17     @(cross(V(in) - vc, 1)) // Goes above vc
18         is_on = 1;
19
20     @(cross(V(in) + vc, -1)) // Goes below -vc
21         is_on = 0;
22
23     V(out) <+ transition(is_on*vout*(V(in) > vth),0,100p);
24
25 end
26
27 endmodule

```

2.2.3 Simplified model for large-scale simulation

To reduce the computational cost of simulating large circuits with complex models requiring multiple bytes of memory to store their state, as well as complex interactions that could cause convergence issues in specific cases, a complete Preisach or Landau model can be replaced with a simplified one. We implemented such a simplified model in Verilog-A, in Listing 2.1.

In this case, there are multiple considerations to take into account:

- Individual circuits should be validated with a realistic model before using a simplified one to simulate medium- to large-scale architectures.
- The simplified model's parameters such as output and threshold voltages should be adjusted to properly represent the realistic model inside the simulated circuit.
- A simplified model such as that given in Listing 2.1 does not (and in general is not intended to) reflect more complex behavior, such as minor cycles or partial polarization.

The simplified model presented in Listing 2.1 is constructed around the idea of a voltage-controlled ($v(in)$) voltage generator ($v(out)$) with a memory (is_on).

To make the model as light as possible, the proposed code makes several assumptions and simplifications:

- the model is asymmetric. It has one input of infinite impedance, and an output with zero impedance. Therefore, a voltage-controlled input is assumed.
- output current and energy are not limited. It is assumed that the output will be connected to a floating node, the steady state voltage of which has been measured using a more complete model.
- V_C is assumed symmetric: $V_C^- = -V_C^+$

- the polarization change is instantaneous upon crossing V_C , although the output has an arbitrary 100 ps transition period, to help with convergence. This low value was chosen to avoid introducing supplementary latency, but could be tuned with latency measurements from characterization.
- it is assumed that the model will be used in a FeFET, as described in subsection 2.4.3: it includes a threshold voltage instead of adding a fixed voltage to the output voltage as would `V(out) <+ transition(V(in)+is_on*deltaV,0,100p)`.

Nevertheless, thanks to its simplicity and stability, it is quite well suited for exploring more complex architectures. The use of a standard `transition` function ensures the continuity of the output signal for simulation stability. This particular model proved valuable during the verification phase of a large-scale circuit discussed in subsection 4.6.4, containing multiple thousands of FeFETs. Using external passive components such as a capacitor in series, or small adjustments to the model as described above, the model can also be used in a broad range of use-cases. Listing 2.1 is best used in series with a transistor gate, as will be discussed in subsection 2.4.3.

2.3 Ferroelectric capacitors

As seen in subsection 2.1.3, ferroelectric materials can be used as high-performance dielectric materials in a regular capacitor structure. The resulting device is called a FeCaps, and can be seen as a regular capacitor with an extra charge storage capacity controlled by the voltage history.

2.3.1 Regular capacitor

The “regular capacitor” regime can be exploited from 0 V to $\pm V_C$. In this voltage interval, the applied voltage is insufficient to repolarize the ferroelectric crystal, which then acts as a regular capacitor dielectric. This capacitor behaves in the same way, regardless of the voltage history or ferroelectric orientation, except for the usually negligible tunneling losses described in section 2.1.3. This makes them well-suited as low-voltage capacitors, including for DRAM designs. A polarization reversal may pre-charge the capacitor, but the polarization history does not noticeably affect the P - V curve close to zero volts.

2.3.2 Non-volatility

A Non-Volatile Memory does not need an external power supply to retain information, in contrast to *volatile* memories. This property makes them well-suited for long-term data storage, as their often worse performance (with respect to volatile memories) is offset by energy savings, and increased reliability against power outages. Volatile memories include DRAM and SRAM. Non-volatile memories include flash memory (common in USB drives, SD cards and Solid-State Disks (SSDs)), magnetoresistive RAM, phase change memories as well as obsolete magnetic-core memory. A more detailed comparison is provided in subsection 2.5.4.

As long as the electric field across the ferroelectric crystal does not reach $\pm E_C$, the internal polarization of the ferroelectric domains remains stable, which makes it a good candidate for a Non-Volatile Memory.

The internal polarization can only be changed when the supply voltage exceeds a certain value, resulting in the electric field across the ferroelectric crystal exceeding the $\pm E_C$ threshold. If the polarization is reversed, the stored charges are released, which leads to a current peak. This behavior is similar to a capacitance increase while crossing the threshold. As the input voltage continues to rise, more domains can be re-polarized, continuing to virtually increase the capacitance until all domains are polarized, or until the voltage is lowered again.

Since the current peak only appears if the polarization is reversed, as illustrated in Figure 2.8, its presence can be used to infer the previous state of the ferroelectric crystal. This is the basic readout principle used by 1T1C memories, which will be studied in section 3.2. Since the read operation sets the memory to a new known state, it is said to be *destructive*, and the previous value needs to be re-written if the stored value needs to be re-read at a later time.

Non-Volatile Memories are one of the most promising applications of ferroelectric materials, and the $]-V_c; +V_c[$ window (in which the stored information is not altered) leaves the door open for future hybrid volatile, capacitor-based **DRAM** and non-volatile, ferroelectric-based memories.

2.3.3 Negative capacitance

As visible in **Figure 2.12**, each ferroelectric domain has an area where the P - V curve is inverted with respect to a normal capacitor: the center area decreases monotonically, instead of increasing.

As the slope of the P - V curve is the capacitance ($C = Q/V = A \cdot P/V$), this zone translates into a negative capacitance, where increasing the voltage decreases the quantity of charge. Such a negative capacitance can be used in series with a regular capacitance, to increase the rate at which the regular capacitance accumulates charge. It has thus been proposed as a mechanism to boost **Metal-Oxide-Semiconductor (MOS)** transistor switching speeds by lowering the subthreshold slope below the theoretical limit of 60 mV/dec for conventional **FETs**, resulting in a device known as **NCFET**.

However, maintaining this regime is extremely difficult in practice:

1. As visible in **Figure 2.12**, the ferroelectric capacitor might need to be put in a pre-determined state before it can operate in the correct region.
2. No other domains should change polarization during the operation of the device; otherwise they will have the opposite effect: this limits the use to the domain with the lowest E_C in the device.
3. The operation range and slope will vary from one device to the other due to variability.
4. A parallel paraelectric capacitor still exists, so their ratio must be adjusted, while balancing size and variability.

2.4 Ferroelectric transistors

2.4.1 FeFET Devices

A **Ferroelectric Field-Effect Transistor (FeFET)** is a **Field Effect Transistor (FET)** in which a ferroelectric layer helps to control the electric field. Most commonly, these are **MOS** transistors with ferroelectric materials integrated inside the gate stack.

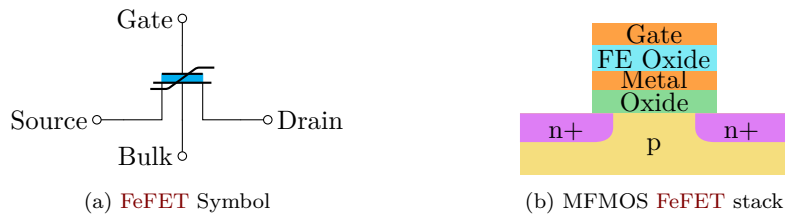


FIGURE 2.18: **FeFET** symbol next to a possible gate stack, that adds a metal and ferroelectric oxide layer to the **MOS** stack.

The main comparative advantage of **FeFETs** against single ferroelectric capacitors is the non-destructive read ability afforded by the use of an extra transistor as an integrated readout mechanism. After polarization reversal, charges are freed from the surface of the ferroelectric layer. Instead of returning to the power supply, they are trapped between the transistor gate and ferroelectric layer, as the two act as capacitors in series. These trapped charges influence the creation of a channel in the transistor: in other words, they shift its threshold voltage, as detailed later in **subsection 4.1.2**.

In turn, this threshold voltage shift allows the stored value to be read non-destructively, while charges remain confined, making the memory non-volatile.

It should be stressed that the output value of the transistor depends on the number of charges trapped near the transistor gate, and not directly on the ferroelectric polarization: the retention mechanism is different (actually closer to that of **flash** memory), and the ferroelectric layer serves in essence as the control mechanism to charge the floating node, as opposed to tunneling current used for **flash** memories. The close proximity of the ferroelectric capacitor to the transistor gate enhances the retention of the trapped charges, but they can still dissipate over time. In such a case, it may be possible to restore them by reversing the polarization again or by **FTJ**-controlled tunneling current, though this has not yet been studied.

The $I_{DS}-V_{GS}$ characteristic of the transistor being modulated by the information stored in the ferroelectric layer, this can be exploited to perform logic operations, as will be detailed in **chapter 4**.

Drawbacks

Despite the very interesting non-destructive read aspect, multiple disadvantages come with using **Front-End of Line (FEoL) FeFET** structures, directly integrated on transistors:

- The most critical issue is the increased voltage required to apply the necessary E_C across the ferroelectric layer. Such voltages can easily exceed of 3 V, which requires compatible transistors capable of handling such relatively high voltages. This might not even be achievable on some process nodes. Special transistors can be necessary as the increased voltage increases the electric field strength across their own gate stack, which might break down the oxide. Circuitry controlling **FeFETs** therefore requires transistors with thicker oxides.
- Strong electric fields can break down the oxide inside the gate stack of the **FeFET** itself, causing additional wear and lower endurance compared to ferroelectric capacitors. Breakdown typically occurs after 10^4 to 10^6 cycles. This is worsened by the usually lower permittivity of the interfacial oxide, which is subjected to most of the field strength, while being thinner [LHS22].
- The capacitance and area ratio of the ferroelectric layer and the transistor is fixed, which makes the design less flexible, unlike with **Back-End of Line (BEoL)** integration of **FeCaps**.

These issues may be mitigated by the use of **BEoL** ferroelectric capacitors connected to the gate of a conventional **FET**, as described in **section 3.3**. The use of an optimized gate stack can also diminish some of the above concerns.

2.4.2 Gate stacks

The ferroelectric layer of a **FeFET** being a dielectric itself, there are multiple ways to integrate it above a **MOS** transistor gate, as illustrated in **Figure 2.19**:

1. **Metal-Ferroelectric-Metal-Oxide-Semiconductor** (MF MOS, MF MIS for Insulator, or more commonly **MF M**) illustrated in **Figure 2.19a**, similar in structure to the **Pseudo-FeFET (PsFeFET)** described in **section 3.3**
2. **Metal-Ferroelectric-Oxide-Semiconductor** (MF OS), illustrated in **Figure 2.19b**
3. **Metal-Ferroelectric-Semiconductor** (MF S), illustrated in **Figure 2.19c**

The first of these gate stacks contains a metal layer, which makes the electric field more homogeneous when applying it across the ferroelectric layer. Indeed, a potential difference could be applied between the transistor gate, and either the bulk (p substrate), source and drain (n-doped areas), or all of them together. Depending on the feasibility of connecting these terminals, there could be an electric gradient across the ferroelectric oxide, or a non vertically-oriented field, thus requiring higher voltages for device polarization [Ni+18]. Moreover, freed charges may not be able to travel across the ferroelectric surface, creating a local depolarization field threatening retention [CL07, p. 29], and having a local effect on the transistor channel. The extra metal layer was also shown to reduce grain (domain) size [Led+21;

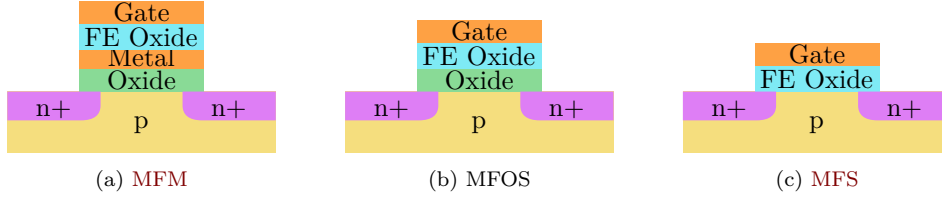


FIGURE 2.19: Cutaway representation of possible **FeFET** gate stacks, progressively removing the intermediate metal layer (2.19b), then the oxide layer (2.19b) from the Metal-Ferroelectric-Metal-Oxide-Semiconductor stack (2.19a).

Leh+21], which is desirable when scaling devices down. **MF MOS** gate stacks are therefore usually preferable to **MF OS**, though they are not always practical to realize, which makes **MF OS** stacks more common.

A second notable difference between **MF MOS** and **MF OS** gate stacks is the position and size of the area where charge screening happens: while charge screening of the lower ferroelectric surface mostly takes place in the transistor channel for metal-less stacks, such screening happens in the intermediate metal layer for the **MFM** stack, owing to the excellent conductivity of metal. This charge displacement changes the local electric charge of the metal layer, but leakage currents may result in the progressive loss of this polarization, more so than in other gate stacks. In turn, this local charge of the metal layer is compensated (screened) in the transistor channel, instead of directly compensating the surface charges of the ferroelectric layer.

Recent developments have been made regarding **MFS** fabrication, notably through **BEoL**-deposited **FeFETs**[**Dut+22**]. Such transistors have a thinner gate stack, which translates to higher fields at equivalent voltages, thus lowering V_C for a given E_C . Moreover, they encounter less charge trapping issues between the ferroelectric and intermediate oxide layer, and are less prone to breaking down the (typically lower- κ , therefore exposed to stronger fields[**LHS22**]) intermediate oxide layer[**Dut+22**]. However, thinner gate oxides will also break down more easily under high electric fields, which limits the achievable voltage gains.

2.4.3 Modeling

A ferroelectric transistor can be modeled by a ferroelectric capacitor connected in series with the gate of a **MOS** transistor. However, such a direct modeling approach assumes that the ferroelectric material is deposited on top of the **MOS** gate metal. While this approach can work for other gate stacks, the model has to be adjusted to compensate for the field orientation, changing the apparent E_C distribution across domains.

Another possible approach is to use a model that acts as a voltage generator in series with the **MOS** transistor, controlled by the state of the **MOS** capacitance[**WA17**]. This approach can better reflect **MFS** stacks, but requires adjusting the **MOS** transistor model, in contrast with a self-contained model. Moreover, a model in series with the **MOS** transistor is able to keep track of the current charging the gate capacitor, making it unclear whether such an approach is necessary.

A more accurate model, especially for non-**MFM** stacks, should be completely integrated with **MOS** transistor models, as this would give access to internal electric fields values and orientations. Existing approaches were deemed satisfactory for the current circuit design needs.

2.5 State-of-the-art on ferroelectric circuits

2.5.1 Ferroelectric Hafnia

While this is not the focus of this document, ferroelectric **HfZrO₂** deposition techniques have progressed greatly over the course of the **3 ϵ FERRO** project[**Bou+19**; **Bou20**; **Fra+19b**], building on previous work, notably at **NaMLab** and **GlobalFoundries**[**SHF19**]. This came with a

better understanding of physical effects leading to the creation of the ferroelectric orthorhombic structure in doped hafnium oxides, leading to improved yields, **Pr**, and other performance criteria such as improved endurance and reduced wake-up effects[Mue+13b; Alc+22].

Modeling

The first step when approaching circuit designs is obtaining good device models. The designs described in this document were first studied with a Landau model[WA17]. While this was sufficient for validating basic circuit functionality, this simplistic model did not account for multiple physical phenomena that could preclude our circuits from working. Moreover, this model was not calibrated to the fabrication processes we were aiming at.

More experimental data and a better understanding of ferroelectric device behavior also led to generalizing the use of improved models, starting with Preisach-based ones[Yin+19], and towards models that represent a wider range of behaviors, including accumulation behavior[Den+20].

2.5.2 Ferroelectric Capacitors and Back-End of Line circuit design

Project partners at CEA-LETI designed then fabricated a 16kbit 1T-1C array[Fra+19a; Fra+21] based on BEoL FeCaps, demonstrating promising performance metrics[Gre+20; Oku+21].

Other uses of FeCaps include neuromorphic computing, where their accumulative switching behavior can emulate the integration mechanism of synapses[Maj22].

Recently, multiple BEoL FeFET designs have surfaced, from the integration of complete FeFET devices in BEoL[Dut+22], to FeFET-like structures similar to the one presented in section 3.3[Ni+18; Leh+21]. This document only covers the latter.

2.5.3 Ferroelectric Field-Effect Transistors-based circuit design

FeFETs are an extremely promising application of ferroelectric material, essentially realizing Floating-Gate MOSFETs (FGMOSs) or flash memory bitcells, but where charge accumulation on the floating gate is controlled via polarization reversal.

This allows controlling the threshold voltage of transistors, as described in subsection 4.1.2 and section 3.3. In turn, this can be leveraged to make an assortment of basic logic gates[Bre+18; Mar+21].

A variety of FeFET-based designs have been proposed recently. Most promising circuits include Content-Addressable Memories (CAMs) and Ternary Content-Addressable Memories (TCAMs), and FeFET arrays for hyperdimensional computing. Large performance and area gains are expected to be achieved for these applications, when compared to regular CMOS implementations: for instance, area can be reduced to a tenth, and the energy-delay product can be improved four-fold for a TCAM design[Yin+19].

FeFETs are well-suited for implementing CAMs, due to their non-destructive readout, and conductance output. This enables the realization of extremely compact CAM arrays, with one bit per FeFET, by chaining their outputs. The TCAM presented in [Yin+19; Ni+19] is based on this principle, using two FeFET to achieve ternary operations. It can be modified to also act as a Random-Access Memory (RAM)[Mar+22], allowing read and write access to individual bits. Multi-level operation can also be leveraged to store multiple bits per FeFET, creating Multi-bit Content-Addressable Memories (MCAMs)[Kaz+21a; Nie+23], with possible applications in hyperdimensional computing[Kaz+21b]. Accuracy can rival that of conventional CMOS implementations, while performance and density are noticeably improved[Kaz+22].

FeFET have also demonstrated neuromorphic behavior for uses in artificial synapses, by leveraging the gradual switching of multiple domains[Mul+17; Jer+17; Maj22].

Alternatively, they are also well-suited for conventional deep neural network architectures, that make extensive use of matrix-vector and multiply-accumulate operations. FeFETs are natural stream processors, which is leveraged in section 4.6 to realize a multiplier. Other architectures have been proposed, notably a matrix-vector multiplier in [Yoo+19] that leverages differential current output to perform the accumulation, although this also requires analog-digital converters.

P-channel Ferroelectric Field-Effect Transistors

Recent progress has been made on p-channel FeFETs fabrication, including on germanium[Zac+22] and silicon-based substrates[Kle+21].

These pave the way for CMOS-type circuits, that eliminate the need for dynamic logic or transresistance circuitry as detailed in section 4.3, provided the off-current leakage remains low.

2.5.4 Comparison with other Non-Volatile Memories

Nowadays, flash is the ubiquitous Non-Volatile Memory solution, used for application from SD Cards to consumer and enterprise SSDs, but also embedded flash (eFlash) in microcontrollers. It is mainly due to the maturity of the technology, its high-density and low cost, as well as ease of fabrication. However, it suffers a variety of drawbacks that limits its usefulness: low write speed, high power requirements, low endurance and vulnerability to radiation. Flash has also been highly optimized, and is uncertain to continue scaling much further.

This resulted in the development of a range of “emerging” Non-Volatile Memory, that are starting to compete with flash in specific applications. A number of candidates with high speed and low power characteristics have emerged: MRAM, PCM, ReRAM, and FeRAM. Among these, HfZrO₂-based FeRAM is expected to achieve excellent endurance and low power consumption, which makes it a promising candidate for replacing flash in embedded applications, especially in normally-off use-cases, to reduce the energy spent during storage and retrieval of processor state.

Desirable characteristics for embedded Non-Volatile Memories (eNVMs) are fast access times, low power consumption, and adaptability to both low-end IoT microcontrollers, and high-end products in a demanding (automotive-grade) environment. High reliability and low-cost are required, including compatibility with manufacturing steps such as solder reflow operations, where pre-programmed memory elements are soldered on circuit boards. Embedded applications typically require smaller amounts of memory, which reduces the density advantage of flash. Two major concerns have thus far delayed widespread adoption of embedded emerging non-volatile memories:

1. lack of confidence in their maturity, manufacturability and reliability,
2. limited high temperature operation range.

While HfZrO₂-based FeRAM exhibits promising compatibility with high-temperature environments, the 3εFERRO project helped demonstrate its ease of integration and reliability. Table 2.2 summarizes the performance characteristics of current-generation HfZrO₂-based FeRAM (both 1T1C and FeFET), and compares them against both mature flash technology, and other emerging technologies. Two key advantages of HfZrO₂-based ferroelectric devices with respect to other emerging Non-Volatile Memory technologies for embedded applications are their very low power consumption and ease of co-integration. HfZrO₂-based ferroelectric memories have demonstrated low-power operation, high endurance (though reduced by destructive read operations in 1T1C structures), high speed, low cost and temperature stability[Fra+21]. Remaining issues relate to charge trapping, ferroelectric loop drift (fatigue, wake-up, imprint) and multiple phenomena related to cycling and data retention.

2.5.5 Design-Space Exploration

The work presented in chapter 5 focuses on Design-Space Exploration (DSE), and leverages previously developed internal tools[Bri21].

DSE of ferroelectric circuits does not often appear in the literature. At the device level, a study[LHS22] attempted to optimize the geometry of FeFET by evaluating the impact of high-k spacers using Technology Computer-Aided Design (TCAD) simulations.

Other studies have focused on the NCFET use-case[SR17; YS17; Pal+18], which is not evaluated in this document. Nevertheless, it is interesting to note that they all leverage the Landau model detailed in subsection 2.2.1.

A DSE of FTJ memory arrays was performed in [Jao+21], with a specific focus on memory selectors and comparison with magnetic tunnel junctions. The results are competitive,

	Flash (mature)	MRAM	PCM	ReRAM	HfZrO ₂ FeRAM FeFET	HfZrO ₂ FeRAM 1T1C
Write energy (pJ/bit)	<200 ^a	0.1 to 20 ^b	~90	~100	<20	<0.1 [Fra+21]
Cell area (μm ²)	0.05	<0.01 [Ike+20]	<0.04	0.1	0.05	0.04
Read access time (ns)	~15 ^c	~1 ^d	~5 (No High Voltage devices)			~4 ^e [Fra+21]
Erase granularity	Page		Bitcell ^f			
Endurance	~10 ⁶ , note ^g	10 ¹⁵ [Gir+21]	5 × 10 ⁵	10 ⁵ , note ^h	10 ⁵ , note ⁱ	10 ¹¹ , note ^j
Retention	~10 years at 55 °C ^k	Energy trade-off ^l	150 °C, note ^m	Energy trade-off	>10 years [Mül+15]	High ⁿ
Solder reflow	Mature	Difficult ^o	Proven	Possible	Likely ^p	Proven [Fra+21]
Extra masks	Many (>10)	Limited (3 to 5)			Few (1 to 3)	
Process flow	Complex		Simple			
New assets vs CMOS	None	New (manufacturable)		High-k materials ^q		

^a~100 pJ/bit for Embedded Select in Trench Memory (eSTM) devices.

^bSub-100 fJ/bit demonstrated for devices scaled down to 11 nm, though with increased error rates[Now+16]. Trade-off with retention.

^cFlash uses high Voltage devices, incurring high energy costs

^dFully logic-compatible (both speed and integration/voltages), could be an alternative to fast SRAM[Gal+19b].

^eDestructive read: 1T1C FeRAM needs Write Back (WB) after read, no high voltage devices

^fBitcell granularity, though architecture-dependent

^gFor flash, cycling frequency proportionally reduces data retention. 10⁶ cycles seems to be the targeted endurance, with up to 10⁷ in some cases. [AG21] quotes 10 years of data retention at 55 °C after 10⁶ writes distributed over 10 years, or 2 years if 10⁶ writes are distributed over 18 months.

^htrade off with bit error rate

ⁱTransistor gates are exposed to high-intensity electric fields, degrading gate oxide. Trapped charges and other mechanisms also limit endurance at 10⁵ to 10⁶ cycles[MG19].

^jhalf for read due to WB

^kData retention is a trade-off with cycling frequency, and decreases with temperature. [AG21] quotes 20 years after 10⁵ cycles spread over 18 months at 55 °C

^lThe main MRAM weakness is the energy-retention trade-off, though it can be tuned for low energy operations, or high-retention. [Now+16] shows a value approaching 10 years at room temperature for scaled cells.

^mcompliant with automotive applications

ⁿTheoretically high, studies[Fra+19a; Fra+21] could not extrapolate retention as the memory window remained open after 10⁴ s at 125 °C

^oStored state is highly susceptible to the solder reflow assembly process, though this can be compensated with higher write currents, shielding, error correction codes and larger devices[Gal+19b; Gal+19a].

^pNot demonstrated experimentally

^qFEoL for FeFET, BEoL otherwise.

TABLE 2.2: Comparison of FeRAM-based memories with other emergent and mature (flash) Non-Volatile Memory technologies. flash is a mature technology, with well-understood performance characteristics and a wide range of commercial suppliers. This table highlights problematic characteristics, as well as major and minor industrialization issues, and targeted performance. Based on project documentation by STMicroelectronics.

but emphasize that increasing the tunneling current causes energy efficiency losses during programming.

A study closer to the present work focused on the impact of the **FeFET** transconductance for deep neural networks[Yoo+19]. However, the scope of that study is narrowed by the use of a sigmoid function as an ad-hoc model for the transconductance linearity, decoupling the study from physical device parameters.

2.5.6 System-level performance evaluation

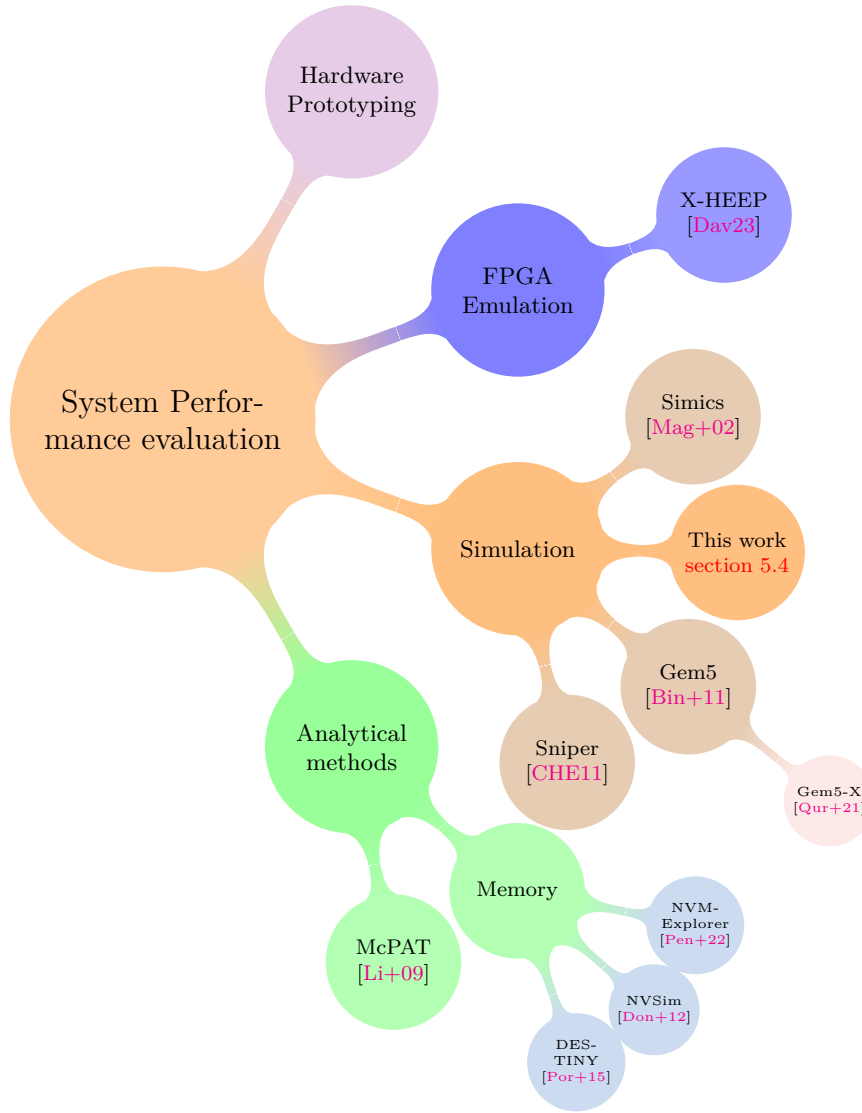


FIGURE 2.20: System-level performance evaluation landscape. While this is not an exhaustive picture, it contextualizes the work presented in [section 5.4](#).

System-level performance evaluation is a crucial aspect of evaluating emerging technologies, and the results can be leveraged to steer device and material-level research efforts more effectively[Nie+23]. While [section 5.4](#) presents a simulator-based approach, multiple alternative solutions exist, as illustrated in [Figure 2.20](#), and described in this section.

Hardware prototyping

The ideal performance evaluation comes from manufacturing a prototype device and measuring its performance. However, this is not an economically viable strategy for testing a

large array of strategies and device configurations. Instead, the accuracy can be approached through models, both hardware and simulation-based.

FPGA-based emulators

Recent open source hardware initiatives, including the freely licensed RISC-V[[Wat16](#)] instruction set, enable faster development cycles by leveraging component reuse. The X-HEEP²[[Dav23](#)] platform targets that use-case by providing facilities for integrating newly-designed accelerators with a RISC-V CPU. The resulting system architecture can then be manufactured, implemented on an **FPGA**, or simulated. **FPGA** emulators offer an interesting trade-off: while less flexible than simulators, they can provide much greater performance, particularly with massively parallel architectures such as **Coarse-Grained Reconfigurable Arrays (CGRAs)**[[Den+22](#)].

Software simulators

Dedicated system simulators such as Gem5[[Bin+11](#)] and derivatives such as Gem5-X[[Qur+21](#)] offer more flexibility, at the cost of runtime speed. Other simulators exist, such as Sniper[[CHE11](#)] and Simics[[Mag+02](#)].

Lastly, analytical methods and tools can estimate the performance of a given architecture without resorting to simulations. These estimates can be less accurate, but are generally faster than simulations, and may not require a complete architectural design, only high-level specifications. Modeling tools such as McPAT[[Li+09](#)] can estimate system-level performance characteristics from technology libraries. Others exist for memory performance, such as DESTINY[[Por+15](#)], NVSim[[Don+12](#)], NVMEexplorer[[Pen+22](#)].

Compiler support and code instrumentation

Compiler support is as a necessary step towards practical, widespread use of non-Von Neumann architectures. While compilers are able to target accelerators from generic code to an extent, for instance through auto-vectorization, this approach is inherently limited, and more optimal use of the hardware can be achieved by explicitly targeting it. Unfortunately, that requires re-writing code and algorithms to take advantage of newly introduced architectural advances, which makes their evaluation more complex. Alternatively, if the code already leverages a programming language or extension that targets coprocessors, such as Halide[[Rag+17](#)], ROCm/HIP[[HIP23](#)], CUDA[[CUDA17](#)], SYCL[[SYCL14](#)], OpenMP[[OMP](#)], etc; the compiler backend may be modified to leverage the new architecture.

The approach described in [subsection 5.4.3](#) is different: the source code of existing algorithms is first modified to print execution traces; these traces are manually converted to the target architecture, and replayed on the simulator. Similar efforts to automatically generate such traces from instrumented code and compilers are underway at CEA-LETI[[Koo+18](#); [Mam+21](#)].

²eXtensible Heterogeneous Energy-Efficient Platform, apologies for the RAS syndrome

Chapter 3

Ferroelectric capacitors-based designs

Contents

3.1 Introduction	53
3.1.1 Back-End of Line technology	53
3.1.2 MAD200 process	54
3.2 1T1C memory bitcell	54
3.2.1 Operation	55
3.2.2 Simulation	58
3.3 FeFET-like structure	58
3.3.1 Description	58
3.3.2 Design	59
3.3.3 Characterization	62
3.3.4 Extension to multi-transistor circuits	64
3.4 Destructive-read TCAM	65
3.4.1 Description	65
3.4.2 Design	67
3.5 2T1C versatile bitcell	68
3.5.1 Description	68
3.5.2 Design	71
3.5.3 Characterization results	74
3.6 Conclusion	76
3.6.1 1T1C memory bitcell	76
3.6.2 Back-End of Line FeFET-like structure	76
3.6.3 Destructive-read TCAM	77
3.6.4 2T1C	77

3.1 Introduction

3.1.1 Back-End of Line technology

Thanks to their compatibility with CMOS design and fabrication processes and the relatively low annealing temperature required of about 450 °C[Bou20, p. 44], doped ferroelectric hafnium oxides (hafnia) such as HfZrO_2 can be deposited above lower metal layers without melting already-deposited transistors and interconnections, as illustrated in Figure 3.1. This is called BEoL deposition, and its current use is limited to Ferroelectric Capacitors (FeCaps).

BEoL deposition can be cost-effective for FeCaps, as capacitor structures are typically larger than logic circuits from lower layers.

BEoL Ferroelectric Capacitors:

- can be designed with a larger feature size, enabling the use of cheaper masks and processes

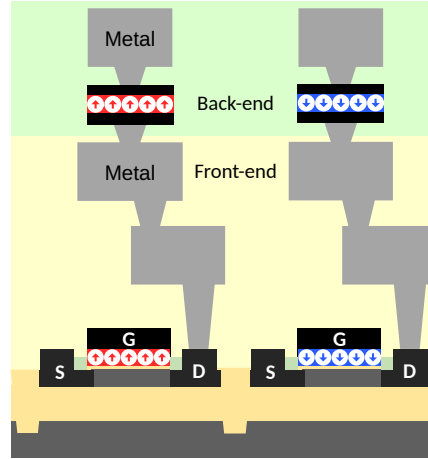


FIGURE 3.1: Illustration of front-end (yellow, bottom) and back-end (green, top) ferroelectric technology. Back-End of Line (BEoL) is deposited after Front-End of Line (FEoL)

- can be higher-capacity (physically larger) without reducing logic density
- are decoupled from transistor sizes, enabling more design options as explained in [section 3.3](#) and [subsection 3.6.4](#)
- are generally of higher quality, leading to enhanced endurance
- are simpler to integrate, as fine handling of oxide interfaces is not required

In the context of this thesis, exploratory FeCap-based logic circuits with full-custom layout circuit designs were submitted as part of a multi-project wafer. These circuits were fabricated with a range of 300 nm, 400 nm and 550 nm diameter ferroelectric capacitors: 2T1C (3.5), “Pseudo-FeFET” (3.3) and TCAM (3.4.1) structures, and will each be detailed in their section.

3.1.2 MAD200 process

The fabrication process used for this work is codenamed **MAD200**, a variant of the HCMOS9A process from **STMicroelectronics**, itself derived from the HCMOS9GP 130 nm technology process.

HCMOS9A stops the HCMOS9GP process after depositing the fourth metal layer (M_4). An **OxRAM** layer is then deposited on top on M_4 by **CEA-LETI**, with an extra metal layer (M_5) deposited above to make contact with the back of the **OxRAM** layer as well as contact pads. The **MAD200** variant replaces that **OxRAM** layer with a HfZrO_2 layer deposited at **CEA-LETI**.

The resulting structure is visible in [Figure 3.2](#) illustrated in [Figure 3.5](#), with the HfZrO_2 layer visible between metal layers M_4 and M_5 .

3.2 1T1C memory bitcell

1T1C stands for “one transistor, one capacitor”, and succinctly describes the memory bitcell pictured in [Circuit 3.1](#). Although this structure is used commonly with **DRAM** arrays, a **FeCap** can be used instead of a regular capacitor in the bitcell, thus making the memory non-volatile. The separation between transistor and capacitor makes this circuit particularly well suited to **BEoL** integration, as larger capacitors can be deposited above control and addressing logic.

This memory cell is one of the simplest ferroelectrics-based circuit. While it was not part of the **MAD200** design contributions, its functionality forms the basis for other circuits, and it was thoroughly explored via simulation. Moreover, **CEA-LETI** contributed a 16 kbit 1T1C RAM array as part of the **3 ϵ FERRO** project, the development of which served as a basis for the **MAD200** process[[Fra+21](#)].

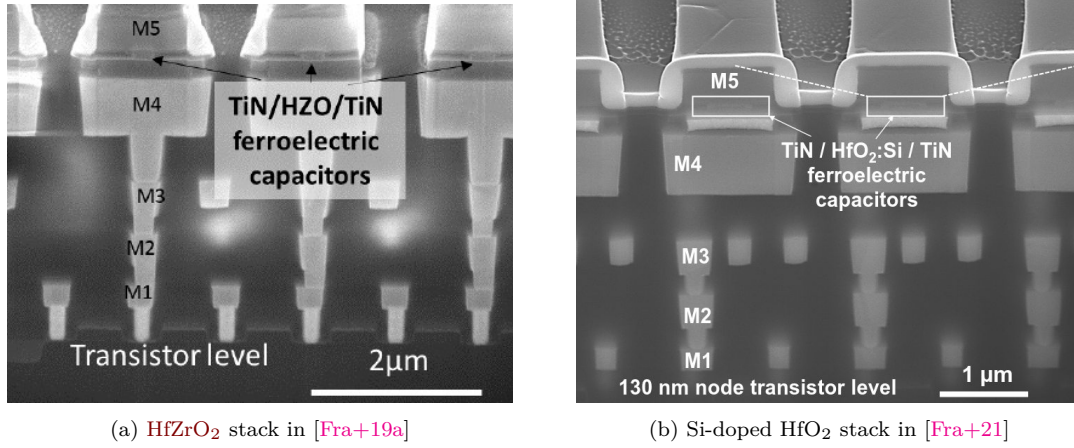
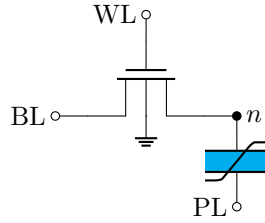


FIGURE 3.2: Electron microscope images of the superior MAD200 layers, including the BEoL ferroelectric layer. Images courtesy of CEA-LETI.



CIRCUIT 3.1: 1T1C bitcell. n is the floating node between the access transistor and the FeCap. In contrast to regular DRAM, the capacitor dielectric is composed of a ferroelectric material, which enables non-volatile operation.

3.2.1 Operation

Bitcell selection and programming

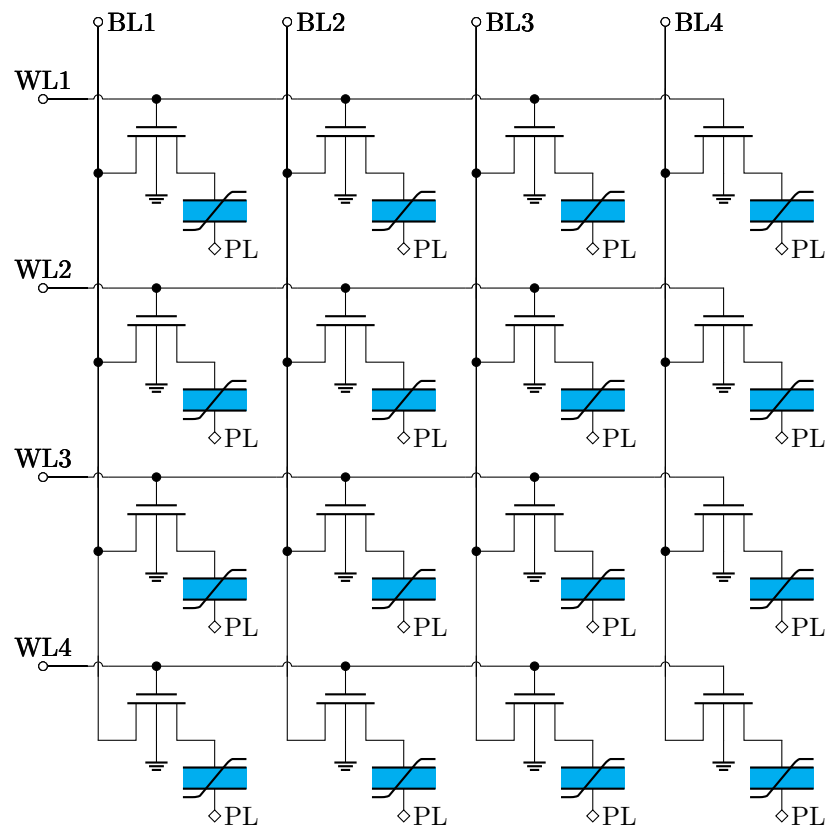
Like DRAM bitcells, this structure is well-suited for dense integration into a memory array such as the one pictured in Circuit 3.2. In these capacitor arrays[Mik+19], a single cell is selected at the intersection of a Word Line (WL) and Bit Line (BL), by activating every access transistor on a given WL, and setting the right BL to the desired voltage.

As visible in Circuit 3.2, the bitcell at the intersection is therefore selected, with the capacitor connected to the active BL. The second terminal of every FeCap is connected to the global Plate Line (PL). A voltage can therefore be applied across the capacitor terminals by controlling $V(BL - PL)$.

To change the polarization of the FeCap, the electric field across the dielectric has to exceed E_C , so the voltage applied between BL and PL should be greater than $\pm V_C$. Two approaches can be taken to reverse the desired polarity, either:

1. directly apply positive or negative voltages to BL or PL while the other remains at a fixed voltage, or
2. apply a positive voltage alternatively to BL and PL while the other line remains grounded.

The first approach requires negative voltages, which can be complex to generate and handle on-chip, but simplifies circuit design, allowing to ground PL, for instance. The second approach needs more control circuitry, in order to supply a positive voltage to either BL or PL, but does not require the use of negative voltages.



CIRCUIT 3.2: 4×4 1T1C array showing four different WL, four BL, and the common PL.

Bitcell readout

In order to read out the stored content of bitcell, and similarly as in **DRAMs**, the programming procedure is repeated with a known voltage of either polarity. This voltage is chosen high enough to reverse the **FeCap** polarization ($|V| > V_C$), while a sense amplifier measures the **BL** or **PL** current. If the polarization was reversed during the new write procedure, a current spike corresponding to the release of trapped charges is registered, and the previous value can be inferred: if a repolarization current is detected, the previously stored value was of the opposite polarity; if no current is detected (besides the paraelectric capacitor current), the polarity applied corresponds to the one stored in the **FeCap**.

However, sensing a current spike means that the previous value was erased, and it therefore needs to be written back to memory if it is to be used later. The read procedure is thus destructive, as is the case with **DRAMs**.

The number of charges released is simply:

$$\Delta Q = 2 \cdot Pr \cdot A_{FE} \quad (3.1)$$

Note that the factor 2 exists because the polarization reversed surface polarity from $\pm Pr$ to the opposite value.

Multi-level memory

Multi-level storage schemes are widely adopted across the industry with other memory technologies, commonly referred to as **MLC**. Using a single memory element to store multiple bits of data by leveraging intermediate states can tremendously increase storage density, at the cost of more complex circuitry, and decreased performance regarding retention time, read and write speed, as well as endurance: individual bits cannot be directly accessed anymore, and need to be read and written to in batches corresponding to the size of each bitcell. In turn, this increases the number of cycles per bitcell, as reading a single bit usually requires writing back the entire bitcell. Conversely, writing a single bit requires (destructively) reading the bitcell before. This strongly discourages random bit-by-bit access patterns, which would drastically decrease the endurance of multi-level cells. Thankfully, that use-case is infrequent, while read and write speeds also benefit from multi-bit operations.

The limiting factor is memory window: as the number of distinct states per bitcell increases, the memory window available for each state decreases. This increases the error rate, reducing retention time and necessitating additional error-correction circuitry. The reduced memory window also makes **MLC** bitcells more prone to fatigue, with the window shrinking with cycling, as more domains fail. Single-level cells could potentially increase the programming voltage in response, to leverage yet-untouched domains, while **MLC** are more likely to start at the maximum permissible voltage.

It is possible to realize multi-level memories with ferroelectric materials, making them more competitive in terms of storage density: polycrystalline ferroelectric material presents a range of coercive fields over the ferroelectric domains, owing to their random orientations, as well as process variability. If a large enough distribution is present, part of the population can be selectively polarized: crystalline domains with a low coercive field can be programmed using a lower programming voltage, or shorter programming pulses.

A single-level (one bit of data per cell) only has two possible orientations, one of them being forced during readout. Therefore, the probability that a value needs to be written back is only $P(WB)_{SLC} = 1/2$: an optimized reading scheme may try to predict the current value in order to avoid write-backs. For n -level cells with 2^n possible states, the write-back probability increases to $P(WB)_{MLC} = 2^{n-1}/2^n$, which makes such prediction schemes less practical, unless highly accurate.

This partially flipped intermediate state is translated into a proportionately lower amount of charges sensed during the readout phase when the domains are forced into a known state. As the quantity of sensed electrical charges is directly linked to the memory window width, it must be large enough to discriminate partial polarization from full polarization reversal. This implies using larger capacitors for multi-level bitcells, which in turns increases the population of ferroelectric domains, making the distribution better-suited to multi-level cells, as this makes the distribution more homogeneous across bitcells.

Indeed, with partial polarization in case of an n -level MLC bitcell, assuming the read pulse targets a full polarization (maximum voltage, leading to $\pm Pr$) from the s^{th} level, a supplementary factor is simply introduced in Equation 3.1:

$$\Delta Q = \frac{s}{2^n} \cdot 2 \cdot Pr \cdot A_{\text{FE}} \quad (3.2)$$

This shows that, as the number n of levels increases, it becomes more difficult to discriminate between individual levels s_i , unless A_{FE} is increased in turn, or technological improvements are made to increase Pr .

To help with MLC read-out, it is also possible to progressively increase the applied voltage, and detect when the first repolarizations occur. This allows reducing the constraints on the sense amplifier and Analog-to-Digital Converter (ADC), making it possible to use 1 bit ADCs, but slows the read operation, and still requires precise control of the applied voltage.

3.2.2 Simulation

Given the simplicity of the circuit, simulations were used to validate the models' functionality and stability. Coupled with design space exploration tools, the simulations allowed the prediction of achievable performance metrics as detailed in section 5.3.1, as well as refining the models by contrasting them against experimental values.

The 1T1C bitcell was simulated with two design kits:

- STMicroelectronics 130 nm MAD200
- GlobalFoundries 28 nm 28SLP

MAD200-based simulations

While no 1T1C cell was specifically designed for the MAD200 production run, the 1T1C bitcell is close to the 2T1C one detailed in section 3.5. The only difference consists of a readout transistor in the 2T1C cell, the use of which is completely optional and which also amounts to an additional parasitic capacitance if left unused, since the bulk of the transistor remains connected to the access transistor in this implementation.

Simulations were performed against specific parameters to evaluate bitcell performance under these sets of conditions, in order to provide performance characteristics for higher-level benchmarking, as detailed in section 5.4.2.

The parameter space was more thoroughly explored in subsection 5.3.1 to determine design compromises.

3.3 FeFET-like structure

3.3.1 Description

FeFETs, as described in chapter 4, are unavailable with BEoL technology, since the ferroelectric oxide is not deposited directly on top of the gate oxide as in a FEoL process, but instead in an upper layer (between the M4 and M5 layers in the MAD200 process). This can be seen as a Metal-Ferroelectric-Metal-Oxide-Semiconductor (MFM) stack[Leh+21], with the second metal layer replaced by a via stack from the transistor gate up to the ferroelectric layer.

The aim of this test structure was to explore the feasibility of replicating FeFET operation by directly connecting the gate contact to the ferroelectric layer through a metal via, as illustrated in figures 3.3, 3.5a and 3.4. The names “Pseudo-FeFET” (PsFeFET), “metal-via FeFET” and “second generation FeFET” have been suggested, as well as “1T1C BEoL FeFET”[Leh+21]. It is also sometimes simply referred to as MFM FeFET or FeMFET.

While this runs contrary to early FeFET developments aiming for close integration[Mue+13a], this had not been investigated previously, and opens up new design possibilities since the physical area of the ferroelectric capacitance can be quite different from that of the transistor gate. It is also anticipated that this could improve endurance, since observed BEoL FeCap endurance (around 10^{12} cycles) far outlasts that of FEoL FeFET (around 10^6 cycles).

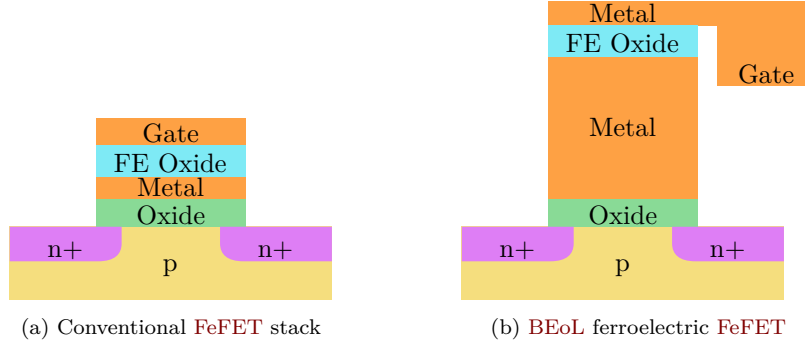
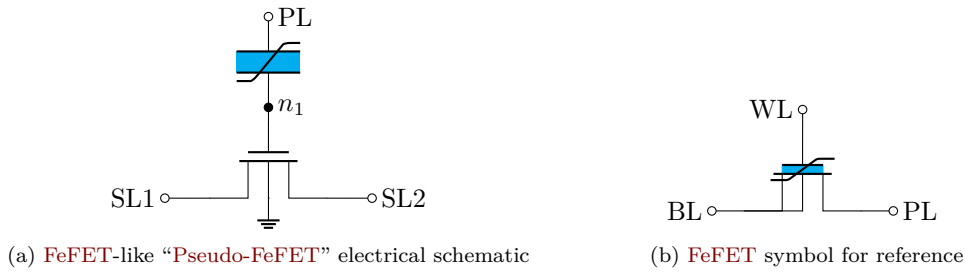


FIGURE 3.3: Cutaway of a conventional FeFET (3.3a), and a proposed “Pseudo-FeFET” stack connecting a conventional n-MOS transistor to a BEoL ferroelectric capacitor (3.3b).

Such structures have subsequently been studied in the literature, and compared to MFM gate stacks[Leh+21].

Unfortunately, these advantages come at the cost of charge trapping risks in the intermediate layer, which can screen the ferroelectric polarization charges or even, in some cases, destroy the oxide. There is also the issue of greater leakage currents from the floating node n_1 visible on Circuit 3.3a. These leakage currents reduce the length of time that a value can be preserved in the FeFET, i.e. its retention. Further, parasitic capacitances introduced by long metal lines also have to be compensated. Once the charges stored in the floating node have dissipated, the value needs to be retrieved again from the ferroelectric layer, which is a destructive operation, and therefore needs to be written back. This differs from regular FeFETs that rely on polarization-screening charges being attracted to the transistor channel, and are thus less likely to be compensated in the intermediate layer: this compensation can still happen in a regular FeFET due to charges being trapped between the ferroelectric and dielectric layer in a metal-ferroelectric-insulator-semiconductor stack, but a larger conductive and charged structure is more likely to attract and retain stray charges.

Lastly, the PsFeFET remains subject to the same drawbacks as the FeFET, such as higher programming voltages than FeCaps. They are also at a greater risk of charge-trapping on the floating node during fabrication, which could threaten the integrity of the transistor gate oxide. Programming voltage concerns can be alleviated by providing access to the floating node (n_1 on Circuit 3.3a), as is the case in the 2T1C structure detailed in section 3.5.



CIRCUIT 3.3: FeFET-like “Pseudo-FeFET” contrasted to a FeFET symbol. PsFeFET should be a drop-in replacement for all intents and purposes, albeit with possibly degraded retention characteristics.

3.3.2 Design

The primary design goal for the PsFeFET was to verify that a functional circuit could be manufactured despite the risks incurred by the gate oxide, to show that device operation is similar to that of a FeFET, and to evaluate its usability in circuits. Secondary goals include the measurement of retention and endurance metrics, leakage current, and the investigation of Logic-in-Memory (LiM) functionality in the 2T1C variant.

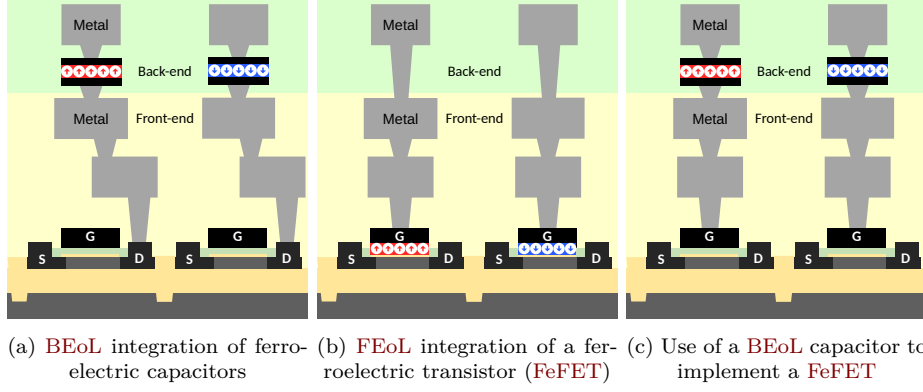


FIGURE 3.4: Cutaway of a PsFeFET (3.4c), next to a BEoL-integrated FeCap (3.4a) and a regular FeFET (3.4b)

As such, the design was approached as a proof-of-concept, and parameters were chosen to maximize operational margins, despite uncertainties such as the remanent ferroelectric polarization, which depends on the proportion of HfZrO_2 that crystallizes into the orthorhombic phase. Test structures of various sizes were designed, as listed in Table 3.1, to compensate for possible process variability, as well as simulation unknowns.

Future iterations could focus on lowering the required operating voltages (in write and read modes), and making the transistor faster by reducing its size.

Capacitance matching

Allowing both the programming and readout of the ferroelectric capacitor requires careful consideration of the various capacitances present in the circuit. The aspects of each operation are somewhat antagonistic, so their relative capacitances need to be considered concurrently:

- To allow reading a value after programming it, the charges released by the FeCap must have a meaningful impact on the floating node potential, therefore suggesting a larger ferroelectric capacitor. This corresponds to maximizing ΔV in Equation 3.5
- To allow programming the FeCap, it must be subjected to a stronger electric field than the transistor, and therefore the transistor should be larger. This corresponds to maximizing V_{fe} in Equation 3.5.

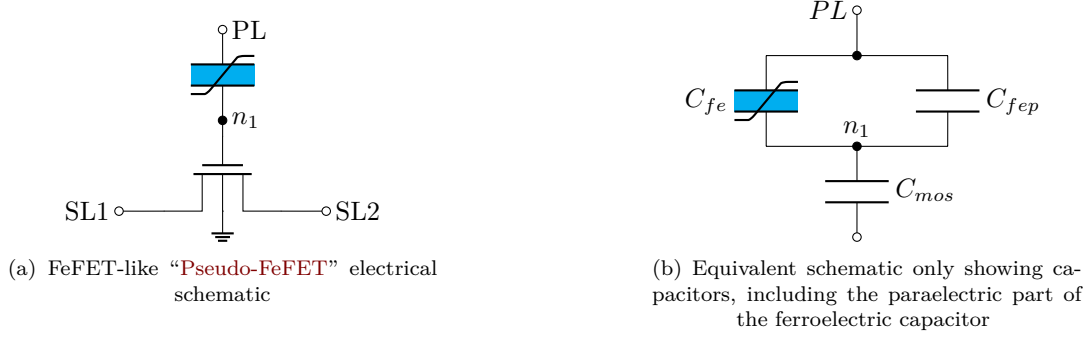
More generally, it is important for the floating node potential to have a meaningful impact on the field across the transistor while reading, and on the same across FeCap while programming. Capacitance matching is also necessary when designing FeFETs, although in this case, capacitances are adjusted via the relative oxide thicknesses and technological parameters rather than through the capacitor and transistor areas.

To allow readout of the capacitor, charges trapped close to the surface of the ferroelectric material (remanent polarization) during programming must have a sufficient effect on the readout transistor when released to make the gate voltage reach above or below the threshold voltage: $\Delta V > 2 \cdot V_{th}$ (assuming no imprint and that the initial, unpolarized state is 0 V, which makes the floating node oscillate between $\pm \Delta V/2$).

To match these values, the capacitor diameter was first set to one of the chosen values of $D_{Cfe} = 300, 400$ or 550 nm. The paraelectric fraction of the ferroelectric capacitances (as illustrated on Circuit 3.4b) was measured during previous manufacturing operations to be comprised between 7 fF and 30 fF for a capacitor size of $2.38 \times 10^{-13} \text{ m}^2$, hence giving between 0.03 and 0.13 F m^{-2} . This allows us to compute the total paraelectric capacitance as C_{fep} . Care was taken to keep the device operational within that range, by adjusting the readout transistor Q_R size so that the gate is guaranteed to reach the threshold voltage when charges are released from the ferroelectric material.

Taking a remanent polarization P_r of 0.19 C m^{-2} , and with the capacitor area $A_{Cfe} = \pi \cdot (\frac{D_{Cfe}}{2})^2$, the number of charges released is:

$$Q_{fe} = 2 \cdot P_r \cdot A_{Cfe}$$



CIRCUIT 3.4: PsFeFET (3.4a), and equivalent schematic (3.4b) showing capacitors, including the MOS capacitance, and decomposing the ferroelectric capacitor into pure paraelectric (C_{fep}) and ferroelectric (C_{fe}) fractions

Note the factor 2, as the remanent polarization switches from P_r to $-P_r$. This way, Q_{fe} charges previously trapped at the ferroelectric surface are released in both the paraelectric fraction C_{fep} of the ferroelectric capacitor and the transistor gate capacitor C_{mos} , producing a ΔV voltage difference at the floating node.

As this voltage difference is applied on the transistor gate, it needs to be important enough to register a change of I_{DS} , such as by crossing the threshold voltage. The gate capacitance C_{mos} can be adjusted accordingly, being proportional to transistor area:

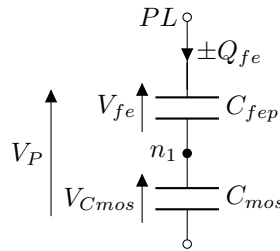
$$\Delta V = \frac{Q_{fe}}{C_{mos} + C_{fep}} \quad (3.3)$$

$$C_{mos} = \frac{Q_{fe}}{\Delta V} - C_{fep}$$

$$C_{mos} = \frac{2 \cdot P_r \cdot A_{C_{fe}}}{\Delta V} - C_{fep}$$

At this point, the smaller A_{fe} (thus smaller C_{mos} proportionally), the higher the voltage, so the better the functionality. However, for programming to be possible, the voltage across the ferroelectric capacitor V_{fe} must cross the coercive value V_c . This needs the MOSFET capacitor to reach a certain size, as when applying a voltage V_P across both capacitors, that value is given by:

$$V_{fe} = V_P \cdot \frac{C_{mos}}{C_{fep} + C_{mos}} \quad (3.4)$$



CIRCUIT 3.5: PsFeFET equivalent circuit from Circuit 3.4b with labeled voltages, and omitting the purely ferroelectric capacitor, that only serves as the source for a charge amount Q_{fe} , which is the same in both series capacitors.

Proof, with the circuit as labeled on Circuit 3.5:

$$V_{fe} = V_P - V_{C_{mos}} = V_P - \frac{Q_{fe}}{C_{mos}} = V_P - \frac{V_P \cdot C_{eq}}{C_{mos}}$$

Matched	TW (μm)	FeCap \varnothing (nm)	TL (nm)	$A_{\text{MOS}}/A_{\text{FE}}$	$C_{\text{MOS}}/C_{\text{FE}}$
Yes	10	300	500	70.7	2.2
	20	400		79.6	2.4
	40	550		84.2	2.6
No	0.5	300		3.5	0.11
	0.5	400		2.0	0.061
	0.5	550		1.1	0.032

TABLE 3.1: Dimensions chosen for matched and unmatched variants of the PsFeFET cell.

This table displays the computed area and capacitance ratio between equivalent FeCap and MOSFET capacitance seen from the floating node, as well as the diameter of the circular FeCap, and the width (TW) and length (TL) of the transistor. Values computed for $\varepsilon_0 \cdot \varepsilon_{r,\text{MOS}}/t_{\text{MOS}} = 3.86 \text{ mF m}^{-2}$ and $\varepsilon_0 \cdot \varepsilon_{r,\text{FE}}/t_{\text{FE}} = 126.3 \text{ mF m}^{-2}$, though compatibility was checked for both $\varepsilon_0 \cdot \varepsilon_{r,\text{FE}}/t_{\text{FE}} = 126.3 \text{ mF m}^{-2}$ and 23.1 mF m^{-2} as contingency. Empty lines in this table have the same value as the closest one above. This table is similar to Table 3.4 as the same dimensions were used for the 2T1C design.

$$\begin{aligned}
 V_{fe} &= V_P - \frac{V_P}{C_{mos}} \cdot \frac{1}{\frac{1}{C_{mos}} + \frac{1}{C_{fep}}} \\
 V_{fe} &= V_P - \frac{V_P \cdot C_{fep}}{C_{mos} + C_{fep}} \\
 V_{fe} &= V_P \cdot \left(1 - \frac{C_{fep}}{C_{mos} + C_{fep}}\right) \\
 V_{fe} &= V_P \cdot \frac{C_{mos}}{C_{fep} + C_{mos}}
 \end{aligned}$$

The quantities to be maximized are both ΔV and V_{fe} :

$$\begin{cases} V_{fe} = V_P \cdot \frac{C_{mos}}{C_{fep} + C_{mos}} \\ \Delta V = \frac{2 \cdot P_r \cdot A_{C_{fep}}}{C_{fep} + C_{mos}} \end{cases} \quad (3.5)$$

The PsFeFET layout as implemented is visible in Circuit 3.6, along with a cross-section and three-dimensional representation of the layer stack in figures 3.5a and 3.5b respectively, though these representations are not accurate in the vertical dimension, neither in distance nor in shape. Corresponding geometries are reported in Table 3.1.

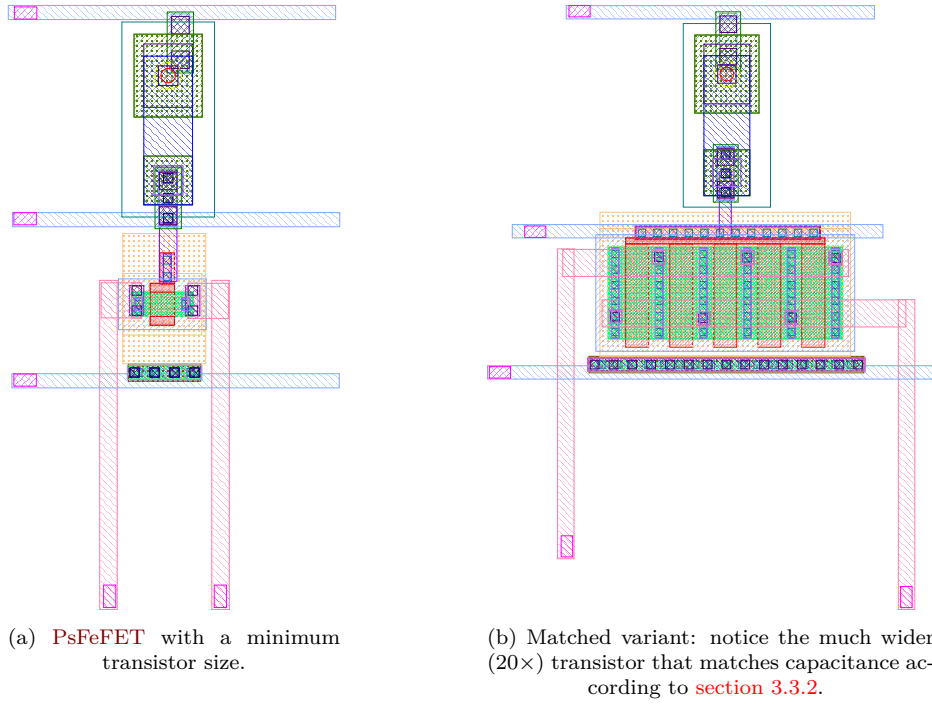
The downside of these structures, as compared to regular FeFETs, is the large floating node, which is a source of charge leakage and parasitic capacitance. This is especially true in this unoptimized implementation: as is visible in Figure 3.5, the topmost metal layer is part of the floating node, while the path could be made shorter by inverting the polarity of the capacitor (vertically mirroring the capacitor in the layout pictured Circuit 3.6). The 400 nm diameter ferroelectric capacitor variant of the 2T1C circuit is different in that regard, and comparing its performance could lead to interesting insights.

Moreover, the design of a follower circuit was initially planned on some circuit variants to provide a direct view of the floating node voltages, though that was later abandoned due to time complexity in fixing antenna design rules errors. This is the reasoning behind an abandoned floating node connection visible in the layout of Circuit 3.6 and Figure 3.5, which probably deteriorates performance further.

3.3.3 Characterization

Protocol and results

Only basic measurements were conducted on the device to date. These were performed as follows:



CIRCUIT 3.6: PsFeFET layout for the MAD200 run, with the capacitor on the left and the transistor on the right. Horizontal lines are the gate, floating node and bulk connection of the FeFET, respectively. Vertical lines are source and drain. Ferroelectric capacitor diameter is 300 nm, and gate width is 500 nm and 10 μm for the unmatched and matched variants, respectively, with a gate length of 500 nm for both, as listed in Table 3.1.

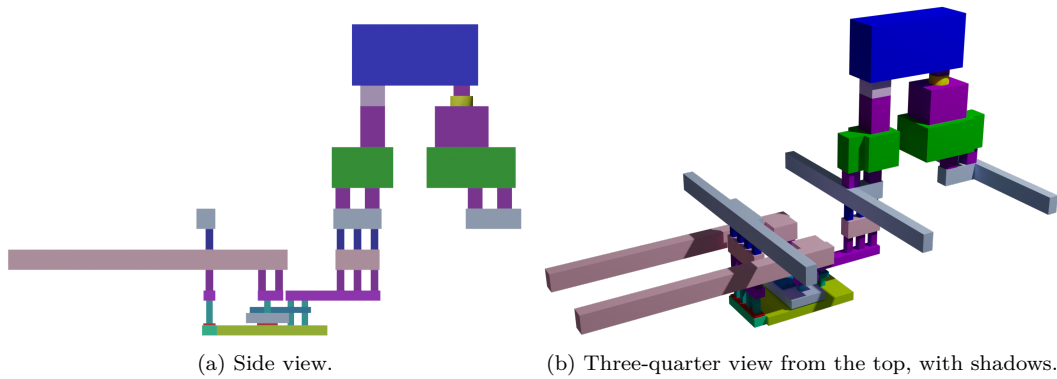


FIGURE 3.5: 3D representation of the layout pictured in Circuit 3.6a. Note that this is a simple extrusion, and while the stacking order is accurate, vertical edges do not accurately represent the physical etching and deposition processes. Moreover, heights are not to scale. Ferroelectric HfZrO_2 is represented by the yellow (●) cylindrical part at the top of the stack, electrically connected between the topmost two metal layers (M_4 , ● and M_5 , ●), with vias (●) as surface electrodes.

1. The capacitor is first erased through the application of a negative voltage on the **PL** terminal. This lowers the floating node potential, which yields a high-threshold state for the **PsFeFET**.
2. The **PsFeFET** gate is then swept from 0.0 V to 1.8 V to extract the $I_D = f(V_G)$ characteristic of the transistor. The ferroelectric oxide is re-polarized during that sweep.
3. The same sweep is then performed in reverse (from 1.8 V to 0.0 V) to provide a reference.

The above was repeated for multiple values of negative voltage pulses, of 1.0 V, 1.5 V, 2.0 V, 2.5 V and 3.0 V. Results are shown on **Figure 3.6**, validating the basic operation of the **PsFeFET**.

During the forward sweep (plotted as a solid line on **Figure 3.6**), the ferroelectric oxide is re-polarized, which releases additional charges into the floating node. This can be seen as lowering the threshold voltage of the **PsFeFET** during the sweep, which results in a virtually steeper sub-threshold slope.

The resulting threshold voltage, after reaching a peak **PsFeFET** gate voltage during forward sweeping, can be observed in the backwards sweep (plotted as a dashed line on **Figure 3.6**), as the ferroelectric oxide polarization is not changed again during backwards sweeping.

The memory window is defined as the difference between the threshold voltages of the erased and programmed states (illustrated with a blue arrow on **Figure 3.6**). The graph shows that the memory window increases with the **Plate Line (PL)** amplitude used to repolarize the device, which was expected. A memory window of up to 0.8 V was extracted during this series of experimental measurements.

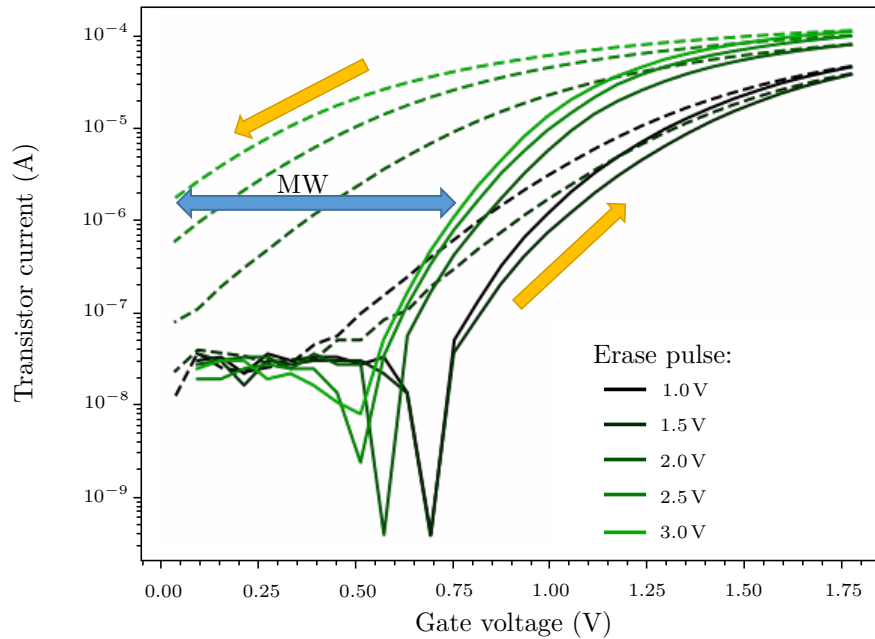




FIGURE 3.6: $I_D = f(V_G)$ characteristic of **PsFeFET** after application of negative-polarity **PL** pulses of different voltages. Forward-sweep (solid line) and backward sweep (dashed line) are indicated with orange  arrows. The memory window is indicated by a blue  arrow. As reported in [Ni+18], V_C appears to be lower than on **FEoL** devices, around 2 V.

3.3.4 Extension to multi-transistor circuits

While this variant was not fabricated, it should be possible to use the same structure for shifting the threshold voltage of multiple transistors at once when the input data is the same.

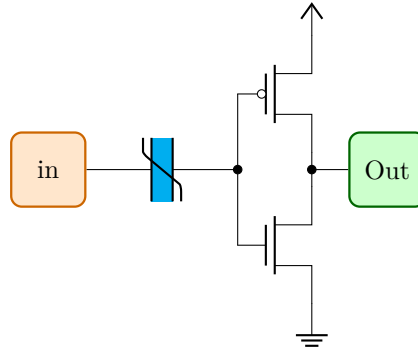
This may be particularly interesting in **CMOS** circuits, where transistors typically share the same input for both n- and p-channel transistors. The capacitance matching process is the

same regardless of the number of transistors; C_{mos} corresponds to the equivalent capacitance of the multiple MOS transistors connected to the same capacitance. This is illustrated in [Circuit 3.7](#) with an inverter circuit. This structure presents the advantages of:

- reducing the number of FeCaps to program, simplifying programming protocols;
- reducing circuit area by reducing the number of FeCaps and simplifying programming circuits;
- increasing circuit density further by dividing the area required for capacitance matching among multiple transistors.

Upon repolarization, the FeCap releases Q_{fe} charges that will affect every transistor, with the floating node voltage raising (or lowering) according to [Equation 3.3](#), inversely proportionally to the equivalent capacitance. This ΔV shifts threshold voltages of p-MOS and n-MOS in the same direction, allowing to boost or lower an input signal above or below the regular CMOS circuit threshold.

It remains possible to size transistors differently (p-MOS being typically larger to offset their lower conductivity), though capacitance matching must be performed after choosing the transistors' relative sizes. Special attention must be paid to the programming phase, as a voltage greater than V_C must be applied across the FeCap. The applied voltage is applied across a capacitive voltage divider, as detailed in [subsection 3.3.2](#). However, unlike in the PsFeFET case, this voltage can be applied across a number of devices. There are two effects to consider when selecting devices through which to present the programming voltage: the damaging impact of electric field across gate dielectrics, and capacitance matching, as a comparatively smaller C_{fe} makes programming easier, from [Equation 3.4](#). Therefore, smaller devices can be spared the stronger electric field, or, perhaps more interestingly, a single, separate larger capacitance could be used only during programming, similar to the access transistor of the 2T1C circuit described in [section 3.5](#).



CIRCUIT 3.7: CMOS-based inverter circuit constructed from a shared Fe-Cap, forming a variable- V_{th} inverter. Considering the ferroelectric polarization as the secondary input, and depending on the choice of threshold voltages, capacitor size and polarization, this circuit may be considered a NAND2 as described in [subsection 4.4.1](#) (either always high or an inverter), a NOR2 (inverter or always low), ignore the input signal (always high or low), or a variation of these (including analog modulation via partial polarization).

3.4 Destructive-read TCAM

3.4.1 Description

Ternary Content-Addressable Memory

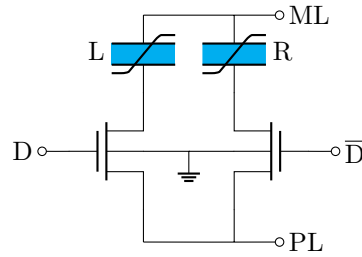
Regular memories store information in an array-like fashion, with each stored value at a chosen position in the array. The memory is then queried using the position index as an address to recall the value.

A **Content-Addressable Memory (CAM)** is a memory that can be queried using the stored values instead of an index. Upon query, the position of the content that corresponds to the queried value is returned. If multiple positions contain that value, all matching indices can be returned, depending on the implementation.

A **Ternary Content-Addressable Memory (TCAM)** allows “wildcard” states to be specified, that will match any queried value. Such states can be specified at bit granularity, and will be represented as X for “don’t care” as a value. This third possible state gives its name to the “ternary” **CAM** (or **TCAM**).

These structures are often designed to be chained in order to perform multi-bit lookups for words of various sizes, and their traditional **CMOS** implementations bear a relatively high transistor count. They are mainly used in networking applications such as routing table storage, but also increasingly in neural network applications.

Operating principle



CIRCUIT 3.8: Destructive-read **TCAM**. It is worth noting that inputs D and \bar{D} are complementary in normal operation mode, but can take the same value for ternary operations

The design presented in **Circuit 3.8** is the result of an attempt at realizing a **FeCap**-based **Ternary Content-Addressable Memory**. As depicted in the figure, it consists of two ferroelectric capacitors, each connected in series with two transistors. The goal is to provide an input signal and find whether it matches the content stored in the capacitor.

This specific implementation works by first storing complementary values in both capacitors. During the look-up phase, a potential difference greater than V_C is applied between **Match Line (ML)** and **PL**. A complemented value is then presented on inputs D and \bar{D} . This activates one of the two transistors, possibly causing the polarization reversal of one of the **FeCaps**, depending on its initial polarization. Polarization reversal can be interpreted in multiple ways (match or not match), but if multiple devices are to be used in parallel, it may be preferable to choose the absence of polarization reversal (and therefore polarization current) as a match, as a complete match will therefore generate no current, instead of an arbitrarily high current value. In that case, the full sequence is as follows (using negative voltages to simplify):

- Initial programming: write A to **FeCap** L , and \bar{A} to **FeCap** R :
- Present V_C on **ML**, and 0 V on **PL**, then apply A and \bar{A} to D and \bar{D} respectively: if A is logic high, L is polarized. If A is logic low, R is polarized. Return D and \bar{D} to 0 .
- Perform the same operation with $-V_C$ on **ML**, and invert the values previously input to D and \bar{D} : the second **FeCap** will be polarized in the opposite direction.
- For matching, the first write operation is performed again with a value B : V_C is applied to **ML**, then B and \bar{B} are applied to D and \bar{D} respectively. A repolarization occurs only if B was different from A . If both D and \bar{D} are presented with a logic low state, no repolarization will occur, providing the “don’t care”, always matching state.

A “don’t care” state can either be stored in the **TCAM** cell, or provided as input data. In either case, the bitcell will always produce a match for that bit. Such a bit can be indicated by simply making sure that none of the **FeCap** will be repolarized, therefore either storing

Saved value			Input value			Match	Comment
L	R	Eq	D	\overline{D}	Eq		
0	1	H	0	1	H	1	Normal operation. Matches.
0	1	H	1	0	L	0	Normal operation. Mismatches.
0	1	H	1	1	\overline{X}	0	Never matches.
0	1	H	0	0	X	1	Ternary operation (“X” lookup). Matches.
1	0	L	0	1	H	0	Normal operation. Mismatches.
1	0	L	1	0	L	1	Normal operation. Matches.
1	0	L	1	1	\overline{X}	0	Never matches.
1	0	L	0	0	X	1	Ternary operation (“X” lookup). Matches.
0	0	\overline{X}	0	1	H	0	Never matches, except for input “X”
0	0	\overline{X}	1	0	L	0	
0	0	\overline{X}	1	1	\overline{X}	0	
0	0	\overline{X}	0	0	X	1	
1	1	X	0	1	H	1	Always matches (“X” stored): both capacitors already flipped
1	1	X	1	0	L	1	
1	1	X	1	1	\overline{X}	1	
1	1	X	0	0	X	1	

TABLE 3.2: Logic states of the destructive **TCAM** circuit. The “Don’t care” (X) state can either be provided as 0 for both inputs, or stored as 1 in both capacitors. An extra state “never match” (\overline{X}) also exists, though it may be of limited use. In this table, a match occurs (“1”) if no polarization reversal occurred, therefore {Left = 0; Right = 1} is equivalent (as shown in the “Eq” columns) to storing or inputting a logic high. Regular **TCAM** operations are highlighted.

the same value in both capacitors, or providing the same input value on D and \overline{D} , below the threshold voltage.

Another use for similar structures is more generic **In-Memory Computing (IMC)**, either as digital or analog form: the Hamming distance between input and stored values can be measured, for instance, by measuring the number of **FeCap** changed. This is done by measuring the total repolarization current, or the number of charges transported during repolarization. Additional weighting could be performed by modulating the amplitude of the initial programming pulse in each capacitor. Multi-level storage in **FeCaps** could also theoretically increase the number of matching bits per **FeCap**.

It is also possible to trade the ternary information for a supplementary matching bit, by avoiding complemented storage and accesses.

The full logic table of the **TCAM** shown in **Circuit 3.8** is given in **Table 3.2**. It can be leveraged to implement either the **TCAM** functionality, or other generic **IMC** functions.

Limitations

Since this structure is based essentially on two parallel 1T1C circuits, it presents the downside of deleting the information upon reading (destructive read), which severely restricts its usefulness compared to **FeFET**-based **TCAMs**[[Yin+19](#); [Ni+19](#)]. As it stores two bits of information and has a single one-bit output, it cannot easily be written back to its previous state (due to the non-complementarity of the stored “X” state).

It can nevertheless be useful in limited situations, such as when accumulating data, then needing to look up a match only once, or in-memory computing operations.

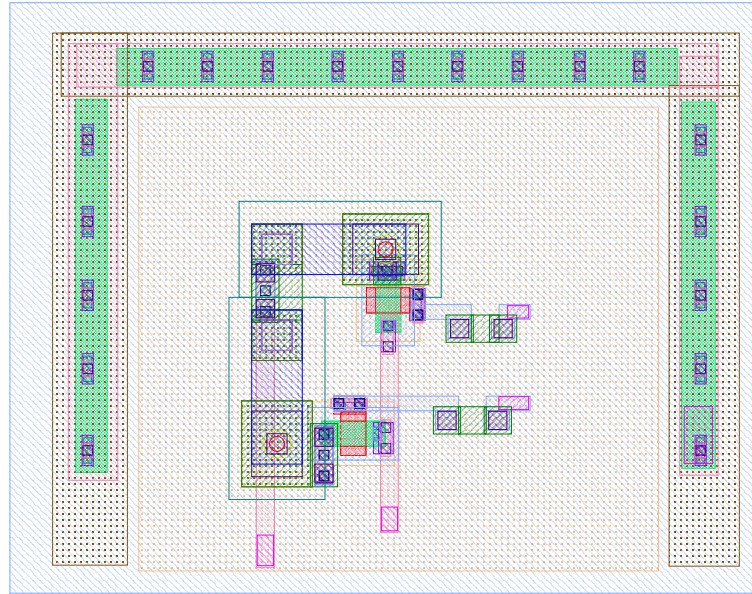
3.4.2 Design

Circuit 3.9 shows the layout for a **TCAM** structure as implemented for the **MAD200** run, while parameters are listed in **Table 3.3**. While transistor were kept at minimum physical dimensions for thick gate oxide devices: $L = W = 500$ nm, multiple **FeCap** diameters have

been fabricated, as well as variants connecting together the **MLs** of two and three **TCAM** cells, respectively. This allows matching up to three bits in the **TCAM** mode, or six in the **CAM** mode (for single-level cells).

FeCap \varnothing	WL	Bits (TCAM)
300	2	1
400	2	1
550	2	1
400	4	2
400	6	3

TABLE 3.3: Destructive **TCAM** FeCap diameters, number of **ML** pads, and bits available for matching in the **TCAM** operation mode. Transistor dimensions are kept at $L = W = 500$ nm. On the first three rows figure one-bit **TCAMs**, for which **PLs** are also connected together due to a limited number of available pads. The last two rows are multi-bit **TCAMs**, made by connecting the **ML** of two and three one-bit **TCAMs**, respectively. These use FeCap diameters of 400 nm.

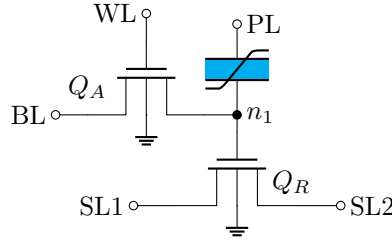


CIRCUIT 3.9: **TCAM** layout. The two large structures are the ferroelectric capacitors, with a vertically descending **ML** and **PL** (pink circle). Access transistors are horizontal connections on the right, connected through a higher metalization level to respect antenna rules. Bulk connections are visible around the structure.

3.5 2T1C versatile bitcell

3.5.1 Description

This structure was first proposed by NaMLab in [Sle+19b; SP21] using a **FeFET** with shorted source and drain terminals to implement the **FeCap** function. However, using a dedicated **FeCap** is a better fit, such that it is of interest to design the structure using a **FeCap** technology such as **MAD200**. This structure is also a good fit for exploring design possibilities enabled by a given fabrication technology, as it allows multiple operating modes for a single device: **FeFET**-like operation, **FTJ**-based operation mode, and a 1T1C **DRAM**-like operation mode; these being, in fact, different ways of reading a value written into the **FeCap**. It also sidesteps



CIRCUIT 3.10: 2T1C bitcell as designed. An access transistor Q_A connects the **FeCap** to the **BL** and is controlled via the **WL**. A second transistor, the read transistor Q_R , is used to monitor the potential of the floating node n_1 during read operation by measuring its source-drain current via **SL1**.

Finally, the **PL** is connected to the second terminal of the **FeCap**.

usual difficulties that can happen when programming a **FeCap** in some devices like the **FeFET**, by providing direct access to both poles of the capacitor.

Programming

Programming the 2T1C is performed identically to the 1T1C case, as described in [subsection 3.2.1](#). Program and erase operations start by activating the access transistor Q_A of [Circuit 3.10](#), which is done by applying a sufficiently high voltage to **WL**. Voltage pulses are then applied across the **FeCap** via **BL** and **PL** in order to switch its internal polarization. In the **FeFET** operation mode, n_1 must be left floating by removing the voltage applied to **BL** before the repolarization, as described in the following subsections.

As with the 1T1C structure, there are multiple ways to handle programming voltages, depending on the constraints. Programming can be performed by:

1. controlling voltages of the selected **BL** ($\pm V_C$) while **PL** is set to ground
2. controlling **PL** directly (setting it to $\pm V_C$), while only the selected **BL** is set to ground
3. in case negative voltages are not available, the selected **BL** can be set to V_C while **PL** is set to ground, and the opposite performed to reverse the polarization

In all three cases, unselected **BLs** are left floating. As a side effect of the polarization scheme, the readout transistor Q_R might be activated. Preserving the gate oxide of this transistor from high voltages may require special consideration if using thin oxide devices.

The test structure was designed to directly connect **PL** to a pad, which makes it easiest to apply negative voltages to this terminal (second case), while other modes remain possible. This was exploited during characterization, where both positive and negative pulses were applied to **PL**. In a more complex circuit, no negative voltages can typically be adopted. Under such conditions, it is necessary to apply a positive pulse to either **PL** or **BL** in order to change the effective pulse polarity seen by the **FeCap** (third case in the list above). Applying large positive voltages to **BL** also requires some consideration: as Q_A is an **n-MOS**, this requires increasing the **WL** voltage, which may be damaging to the access transistor. This could be mitigated by employing an additional **p-MOS** transistor to form a complete transmission gate, at the cost of increased area.

Reading as **FTJ** – Non-destructive read

After pre-charging the floating node using the access capacitor, charges will start leaking through the **FeCap** by tunneling effect, as introduced in [section 2.1.3](#). The intensity of this leakage current depends on the tunneling distance, which depends on the charge screening area length within both **FeCap** electrodes. If that charge screening length is asymmetrical, as would be caused by different materials, the effective tunneling distance is modulated by the ferroelectric polarization[FS19; Jao+21; Maj22]. In turn, this affects the floating node voltage decay rate, which is directly sensed by the reading capacitor Q_R , as the floating node n_1 voltage affects $I_{SL1-SL2}$. Capacitance matching between the read transistor Q_R and the **FeCap** is not crucial in this case, as capacitance will mostly affect the decay rate, so it only

has to remain consistent across a memory array, which is more likely to be achieved with larger devices.

FTJ devices typically use thin-oxide **FeCaps** as well as an additional paraelectric layer on one side (in part to compensate charge trapping issues[PLH21, p. 5]), or two different electrode metals to maximize the tunneling current asymmetry. It was shown that using a semiconductor at one of the electrodes had further potential to increase the asymmetry of the response[GB14; Maj+18; Maj22, pp. 10]. Increasing the area of Q_R will allow the floating node n_1 to retain its potential over a longer period, allowing longer measurements, while possibly making the memory slower: sensing the polarization state can either be performed by measuring the decay rate of Q_R 's I_{DS} , or by sensing the conductivity after a predetermined delay.

Reading as **DRAM**, or 1T1C – Destructive read

The **DRAM** operation mode is almost identical to that of the 1T1C cell, as described in section 3.2, with the addition of a readout transistor connected to the intermediate node. In this operation mode, a voltage is pre-charged on the floating node before applying a different voltage to the opposite capacitor terminal. The voltage difference is greater than the coercive voltage, which changes the ferroelectric polarization if it was different. That polarization reversal is registered by a change in the number of charges in the floating node, which creates a voltage difference sensed by the readout transistor. Matching the capacitance between the floating node capacitance and the charges released when flipping the capacitor polarization is absolutely critical, as it will directly influence the voltage sensed by the readout transistor: the number of charges added or subtracted when reversing the ferroelectric polarization must allow the readout transistor to cross the threshold voltage (i.e. switch state), or at least register a visible change in its drain current.

This *read* procedure is identical to the *writing* procedure of the **FeFET** operation mode.

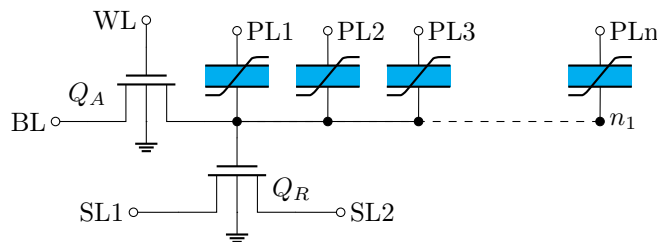
Reading as **FeFET** – Non-destructive read

The 2T1C in **FeFET** operation mode works similarly to the **PsFeFET** device from section 3.3 or that of an actual **FeFET**, but the access transistor allows improved control of the programming phase by allowing to precharge the floating node directly, thus more accurately controlling the voltage across the capacitor during the programming phase. That direct control lowers voltage requirements, lowering the intensity of electric fields applied to the transistor gate oxide, allowing for thinner oxides and preserving the dielectric against electrical breakdown.

This cell can also be programmed without using the access transistor, provided capacitance matching allows the floating node to reach high enough voltages to reprogram the ferroelectric oxide. That is the case for circuits that were designed as part of the **MAD200** run.

It is important to note that to allow a **FeFET**-type read, polarization reversal during the programming phase must happen while the node is floating; otherwise the **FeFET** threshold voltage will not be shifted by the released charges: in this operation mode, the actual non-volatile state stored and read out is the potential of the floating node n_1 . This operating mode therefore requires a different writing procedure.

2T-nC



CIRCUIT 3.11: 2TnC cell, a multi-capacitor variant of the 2T1C cell in Circuit 3.10.

A 2TnC variant was also investigated, which connects multiple **FeCaps** to the floating node instead of a single one, as pictured **Circuit 3.11**. This enables finer tuning of the ferroelectric and gate capacitance, which is critical for some operating modes, such as writing operations for the **FeFET** (non-destructive read) mode. With this circuit design, tuning can be performed after fabrication by deciding to use a higher or smaller number of capacitors and connecting the required number of PL terminals together. Secondly, by writing different values to the **FeCaps**, it is possible to realize in-memory computations[**Sle+19a**; **Rav+19**] such as hamming distance measurements by simultaneously writing to multiple capacitors in **FeFET** or **DRAM** mode, or boolean operations with data stored in each **FeCap**, in either **FTJ** or **DRAM** mode.

This structure also allows the investigation of multi-level memory cells without requiring precise control over programming timing or voltage: in practice, programming a single capacitor is equivalent to polarizing a subset of the domains of the equivalent **FeCap**. Indeed, as shown in **Table C.1** and **Table C.2**, simultaneously selecting multiple **FeCaps** allows addressing and repolarizing a fraction of the total area (possibly over multiple programming passes). This allows serially programming multiple **FeCap** to act as a single, virtual **MLC** during read operations, allowing far more deterministic control over the state stored in the virtual **MLC**. This could also allow the use of smaller **FeCaps**, towards single-domain ones, especially if a mechanism is implemented to replace defective capacitors, or correct the output of a data block in case a capacitor is found faulty. In turn, the transfer function (or transconductance) of the bitcell, including its linearity, can be tuned by changing the selection of capacitor sizes in the array. This could have applications in the field of neuromorphic computing[**Yoo+19**]. Such a fine-grained control requires additional addressing circuitry, which may make it uninteresting, unless coupled with space-efficient structures such as crossbar arrays. Likewise, while it is theoretically possible to simultaneously write to (and therefore read) different capacitors, the added control complexity may make it irrelevant.

Finally, it is interesting to note that this structure is a variant of the **PsFeFET** cell from **section 3.3** (with an additional access transistor) where capacitance matching can be dynamically tuned, making it easier to reprogram the ferroelectric (when selecting a single **FeCap**, applying most of the voltage across it), while being sensitive enough to the floating node polarization and input signal (when providing it on multiple **FeCaps** at once).

Destructive **TCAM** emulation

The 2TnC cell can be used as a substitute for the **TCAM** cell described in **subsection 3.4.1**. In this case, **ML** is connected to the floating node n_1 , and is accessible as **BL** when Q_A is conducting. Contrary to **Circuit 3.8**, **PL** is not shared with an access transistor to address individual capacitors: instead of generating signals for **D** and $\bar{\mathbf{D}}$, each capacitor is accessed through its own **PL** during both the programming and matching phases. That possibility was not envisioned before fabricating the design, and the 2TnC design may also encounter less parasitic capacitance from the **ML** pad, which corresponds to the floating node n_1 in the 2TnC design.

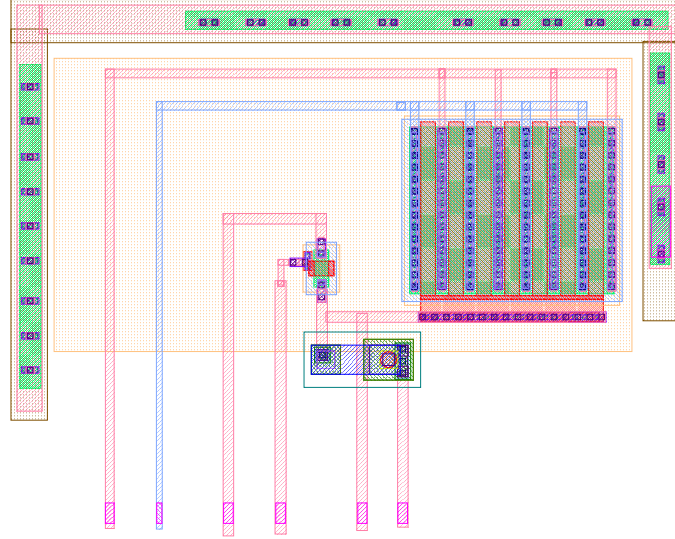
3.5.2 Design

The circuit was realized using the **MAD200 Design Kit (DK)**. The goal was to validate the circuit functionality, as well as evaluate the feasibility of each operating mode with the fabrication technology.

Capacitance matching for **DRAM** operation with destructive read

Capacitance matching is key to the design of the memory cell, in order to enable the **DRAM** operation mode, and more generally to enable the use of the readout transistor.

As the cell is very similar to the **FeFET**-like structure from **section 3.3**, the capacitance needs to be matched as detailed in **section 3.3.2**. It should be mentioned that the access transistor relaxes the constraints on capacitance matching, as the floating node can be pre-charged to a certain value, and it is not necessary to apply a voltage across the transistor stack.



CIRCUIT 3.12: 2T1C layout. Visible on the left is the access transistor, on the right the capacitance-matched read transistor, and at the bottom, the **FeCap** (red circle inside the squares). Vertical connections are, in order: Source and drain, **BL**, **WL**, floating node (used for 2TnC), and **PL**. Bulk connection is visible around the structure. The **FeCap** displayed above has a diameter of 550 nm, and is matched with a transistor of $40\ \mu\text{m}$ per 500 nm, which corresponds to area ratio of 84.2 for a capacitance ratio of 2.6, both in favor of the transistor, according to Table 3.4.

Nevertheless, the greater the voltage swing on the floating node n_1 when flipping the polarization, the easier the readout is, particularly as **BL** is directly connected to a high-capacitance pad, which makes accessing it easier. Moreover, having two similar circuits eases comparisons and diagnostics. It was therefore decided to use the same capacitor and transistor sizes as in section 3.3.

The access transistor Q_A has a minimal impact on floating node capacitance, and was therefore left at $W = 500\ \text{nm}$ and $L = 500\ \text{nm}$, this being the minimum size for a high- V_{th} transistor, capable of sustaining the high voltage operations required by the ferroelectric material, which was the **DK**'s minimum size for transistors capable of sustaining coercive voltages on their gate. Though these transistors can avoid being subjected to such voltages, it is necessary for the **FeFET** programming operation, and the relatively high length is expected to help with reducing current leakage at the floating node.

Circuit 3.12 shows the layout for a 2T1C structure as implemented, for a capacitor diameter of 550 nm, and a transistor geometry of $W = 40\ \mu\text{m} \times L = 500\ \text{nm}$, and a **FeCap** with about 40% the capacitance of the transistor gate, according to Table 3.4.

Alternatively, the access transistor Q_A can be used to precharge the floating node, compensating capacitance mismatches with additional charges.

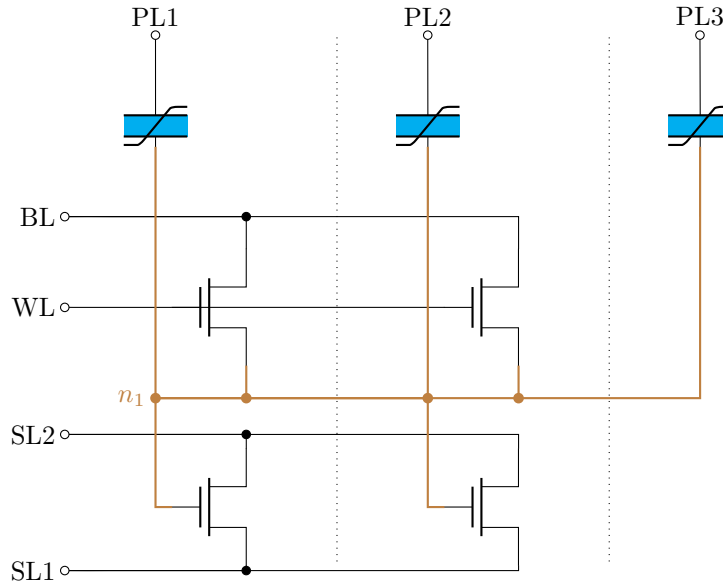
2T-nC

To avoid performing capacitance matching and Design Rules Check (DRC) on the 2TnC structures, the previous designs were re-used, and the floating nodes connected, as shown in Circuit 3.13. As a side effect, the floating node has a much higher capacitance, as well as higher current leakage. Having multiple capacitors connected in parallel to multiple pads allows using a subset if the ferroelectric capacitance were to be too high compared to the transistors. Considering this, a large (550 nm diameter) capacitor was added to every 2TnC structure to allow it to be adjusted in the opposite direction. The available ranges for capacitance matching are listed in Table C.1 and Table C.2 for the 2T4C and 2T5C structures, respectively.

Design	TW (μm)	FeCap \varnothing (nm)	TL (nm)	$A_{\text{MOS}}/A_{\text{FE}}$	$C_{\text{MOS}}/C_{\text{FE}}$
2T1C	10	300	500	70.7	2.2
	20	400		79.6	2.4
	40	550		84.2	2.6
2T4C	70	924		52.1	1.6
		300		495.1	15.1
2T5C		934		51.1	1.6
		300		495.1	15.1

TABLE 3.4: Dimensions chosen for capacitance matching of the 2T1C cell and the 4&5T variants. The multi-FeCap variants are here represented with both their minimum and maximum available capacitance ratios, with equivalent diameters corresponding to the maximum and minimum areas (0 excluded) from Table C.1 and Table C.2.

This table displays the computed area and capacitance ratio between equivalent FeCap and MOSFET capacitance from the floating node, as well as the diameter of the circular FeCap, and the width (TW) and length (TL) of the readout transistor Q_R . Values computed for $\varepsilon_0 \cdot \varepsilon_r \cdot \text{MOS} / t_{\text{MOS}} = 3.86 \text{ mF m}^{-2}$ and $\varepsilon_0 \cdot \varepsilon_r \cdot \text{FE} / t_{\text{FE}} = 126.3 \text{ mF m}^{-2}$, though compatibility was checked for both $\varepsilon_0 \cdot \varepsilon_r \cdot \text{FE} / t_{\text{FE}} = 126.3 \text{ mF m}^{-2}$ and 23.1 mF m^{-2} . Empty lines in this table have the same value as the closest one above. The first lines are similar to Table 3.1 as the PsFeFET dimensions were reused for the 2T1C design.



CIRCUIT 3.13: Example implementation of a 2T-3C structure, similar to the fabricated 2T4C and 2T5C versions. Two 2T-1C structures are joined together, plus one single FeCap. The floating node n_1 has been colored in brown ● for readability.

3.5.3 Characterization results

Preliminary investigation of the device performance was carried out at NaMLab, with further investigation planned depending on availability. The results presented here focus on the 1T1C operation mode (as presented in section 3.5.1), which is less sensitive to charge leakage.

Reference I_{DS} — V_{GS} plot for Q_R

To provide a reference, a voltage sweep is first performed on the readout transistor Q_R to plot its characteristic: the access transistor Q_A is enabled by applying a large voltage (higher than subsequent BL voltages) to WL. This provides direct access to the Q_R transistor gate from BL. A 100 mV voltage is applied between SL1 and SL2, while the current between SL1 and SL2 is measured during the BL sweep from 0 V to 1.75 V.

Figure 3.7 shows the resulting $I_D = f(V_G)$ characteristic of the readout transistor Q_R , after multiple programming and erasing pulses. The I_{DS} — V_{GS} plot has two purposes:

1. Allow the floating node n_1 voltage to be inferred from the measured current
2. Show that the readout transistor behavior is independent of the FeCap state.

Characterization protocol

The following protocol is repeated multiple times to study the impact of different programming voltages, and to validate the basic memory functionality of the system:

1. An erasing (e, here chosen negative) pulse is performed to set the ferroelectric transistor to a known state.
2. A programming (p, chosen positive) pulse of a given voltage is applied.
3. The same I_D measurement is then performed while applying a final read pulse, with the access transistor Q_A disabled.

The resulting evolution of Q_R 's drain-source current during the read pulse is plotted on the right side of Figure 3.7, up to 500 μ s before and after the pulse.

A programming pulse is defined as $V_{PL} - V_{n_1} > V_C$, or equivalently, *only when* Q_A is enabled as $V_{PL} - V_{BL} > V_C$. Likewise, an erase pulse is performed when $V_{PL} - V_{BL} < -V_C$, with V_C representing the Coercive Voltage.

When Q_A is disabled, however, in addition to the PL voltage, the voltage drop is determined mainly by the capacitive voltage divider between the FeCap and the gate capacitance of the read transistor Q_R , hence the capacitance matching considerations of section 3.3.2.

To reach the coercive voltage during the readout phase, the internal node n_1 is pre-charged by activating the access transistor Q_A via WL, applying a suitable voltage to BL, and then switching off the access transistor Q_A . A voltage pulse is then applied between BL and PL.

This results in a voltage change across the FeCap. If that voltage crosses a coercive value while the previously stored value was the opposite one, the polarization is reversed, changing the electrical potential at the surface of the capacitor. In turn, this frees or captures more charges, resulting in a change of potential at the floating node n_1 .

The $V_{BL} - V_{PL}$ voltage pulse has to be chosen in such a way that the voltage drop over the FeCap is large enough to switch it ($> V_c$)

Results and interpretation

Figure 3.7 shows the measurement results. On the left-hand side, the $I_d = f(V_g)$ (I_{DS} — V_{GS}) characteristic of Q_R is shown as characterized beforehand by sweeping Q_R 's gate voltage V_g via the BL while the access transistor Q_A was switched on. The right-hand side shows the evolution of the source-drain current over a 1 μ s window centered on the readout pulse applied to PL. Current values, when taken together with the I_{DS} — V_{GS} characteristic, allow determination of the internal floating node n_1 voltage, as illustrated by the red dashed lines. Multiple current levels can readily be seen on the graph after the programming pulse, each corresponding to a different voltage used for the programming pulse (0 V, 1 V, 2 V, 3 V and 4 V). This demonstrates both proper read-out of stored information, and the capacity to

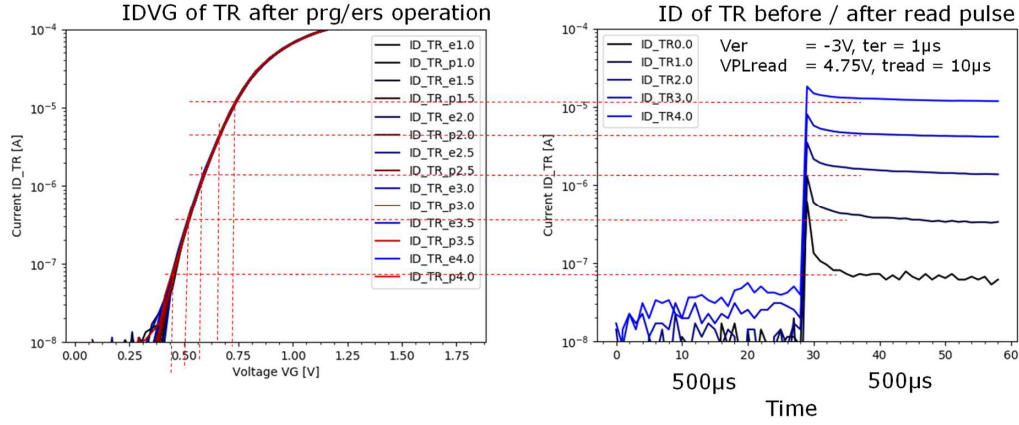


FIGURE 3.7: I_{DS} — V_{GS} characteristic of 2T1C readout transistor, and measured current during read pulse after erase pulse of $V_{er} = -3V$. The graphic on the right shows multiple distinct read current values after programming pulses of 0 V, 1 V, 2 V, 3 V and 4 V, that can be traced back to multiple floating node potentials. This also shows that multi-level memory operation is possible.

The measurements were performed on a single 300 nm diameter circular **Fe-Cap**. The transistor characteristic is plotted for programming ($ID_TR_p^*$) and erase ($ID_TR_e^*$) pulses of 1 μs and 1 V to 4 V; while the read pulse is 10 μs long and 4.75 V high. Current is measured under a 100 mV potential.

discriminate between multiple **Pr** values, thus enabling multi-bit storage (**Multi-Level Cell (MLC)**).

A small current increase was measured after each reading pulse, even during the readout phase of the 0 V programming voltage. That last programming voltage should not change the internal polarization of the **FeCap**, therefore leaving it in the same state as after the erase pulse. The readout pulse, which has the same effect on the **FeCap** as an erase pulse, should therefore not have a measurable impact on the measured current.

This indicates a possible impact of leakage currents through the gate of Q_R or through the **FeCap** during read operations, or the influence of short-term retention effects. This will be the topic of further investigations.

Switching dynamics investigation

In order to better understand the relation between voltage-controlled and pulse width-controlled switching, kinetic measurements were performed. Indeed, both pulse parameters (voltage, width), have an impact on ferroelectric switching, and the objective of this study was to quantify this in more detail. The results are summarized in **Figure 3.8**. As expected, **Pr** clearly depends on both the programming pulse time and voltage, and increasing either leads to a higher **remanent polarization**. A full quantitative analysis of the time-voltage relationship will be the object of further studies, including comparing this circuit with a single capacitor in order to determine whether the 2T1C cell alters switching kinetics.

In order to eliminate cell-to-cell variability, especially for the storage of multiple polarization levels in multi-bit storage, target programming algorithms are typically used. Multiple program and verify steps that require complex pulsing schemes[Zho+20], e.g., variable pulse voltage amplitudes or longer pulse widths are required. This can increase programming operation times, power consumption, and the complexity of circuit architectures.

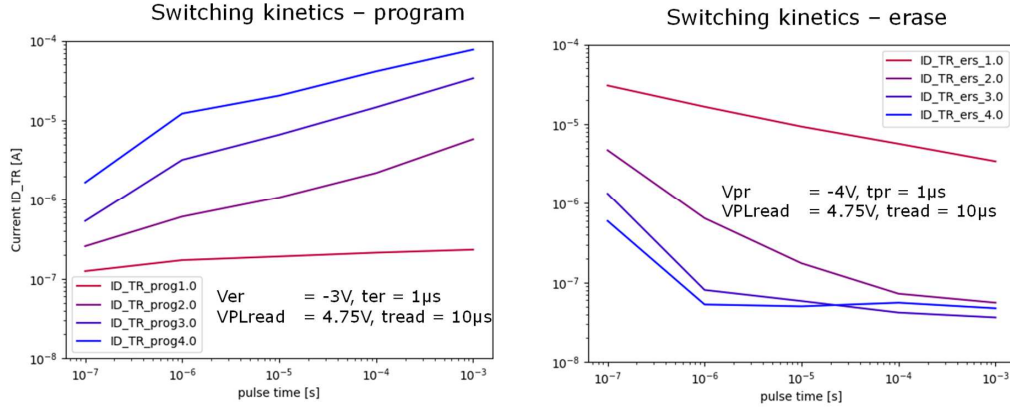


FIGURE 3.8: Measured current of read transistor Q_R after application of different erase-and-program pulses with varying pulse widths (x-axis) and varying pulse amplitudes (legend) using the same 10 μ s, 4.75 V read pulses as in Figure 3.7.

3.6 Conclusion

3.6.1 1T1C memory bitcell

While 1T1C bitcells can only be read destructively, they offer promising performance characteristics, with low-energy, low-latency read and write operations. Such memory arrays have been shown to operate down to 4 ns and 100 fJ bit⁻¹ [Fra+21]. This is in part due to having direct access to both FeCap electrodes for read and programming operations, which lowers voltage requirements compared to a FeFET. With no transistor gate being stressed with high voltages during operation also increases the endurance of the device, though endurance can be traded for higher performance, from 10¹⁵ cycles at 2 V [Oku+21] to 10⁷ cycles at 4 V [Fra+21].

They also have excellent retention characteristics, thanks to directly using the ferroelectric polarization as a memory mechanism.

Due to their simple structure, they are a good target for validating ferroelectric models and simulation approaches. This also makes them a frequent object of study in the scientific literature, with the associated experimental data a possible calibration source for models.

These benefits make HfZrO₂-based 1T1C RAM arrays promising as a future memory technology, and a possible alternative to flash.

3.6.2 Back-End of Line FeFET-like structure

The PsFeFET structure shows promising potential, enabling the use of FeFETs without modification to FEOl processes, thereby making FeFETs available as an add-on technology to existing processes and technological nodes. More extensive characterization activities will be performed on the fabricated design, to better understand their performance relative to FEOl FeFETs. Like the 2T1C in FeFET operation mode, the cell is expected to have degraded retention characteristics, losing the state stored on the floating transistor gate faster than FEOl FeFETs due to additional current leakage sources.

Nevertheless, PsFeFETs are more flexible than FEOl FeFETs, both from a fabrication perspective, as they do not require extensive changes to existing processes, and from a designer perspective, as they can be sized independently of the transistor gate. Moreover, as a MFM gate stack, they inherit better performance characteristics, such as lower programming voltages, and higher endurance thanks to having less charge trapping issues at the interface. Indeed, charges are more mobile on the surface of metal electrodes than at a ferroelectric-insulator interface.

These advantages also lead to a similar structure being investigated by multiple groups [Ni+18; Leh+21], under various names, as described in section 3.3.

3.6.3 Destructive-read TCAM

This circuit was originally envisioned as a **TCAM**, though its destructive operation mode makes it ill-suited for the traditional **TCAM** use-case: frequent searches in a memory array. A router would, for instance, perform such a search for every incoming packet, attempting to find matches for the destination address in the routing table. This use-case would require refreshing part of the **CAM** array with data from an external memory on every access: due to the ternary state, a self-refresh (or **WB**) mode is not possible, as the output bit does not necessarily reflect the stored content. Moreover, it is not possible to reconstruct multiple bits from a single output bit in case the cell is used for matching multi-bit patterns.

This use-case is better served by **TCAMs** that can be read non-destructively, typically **FeFET**-based[Yin+19; Ni+19].

The cell may also be useful for computation, including hamming distance computation, though the 2T1C cell can be leveraged for that purpose.

3.6.4 2T1C

A fully functional 2T1C cell featuring a **BEoL** integrated ferroelectric **HfZrO₂**-based capacitor was demonstrated for the first time. Characterization work was conducted on the 300 nm diameter capacitor variant, the smallest one manufactured during this work. Cells that have been investigated display the expected behavior, besides some interesting differences that can be attributed to leakage currents. Moreover, the 2T1C cell allows the direct experimental characterization of small, scaled single capacitors.

More complex 2TnC cells featuring up to four **FeCaps** within one single cell have been realized as well, allowing simultaneous read and write operation of different capacitors, which enables basic logic functionality for future proof-of-concepts.

2T1C and 2TnC are versatile structures that can emulate a number of other circuits, including partially replicating destructive **TCAM** and **PsFeFET** functionality. It is therefore expected that these cells will continue to be investigated, and will likely serve as proof-of-concepts to demonstrate future circuits. Some important characterization work remains to be done on these cells, including studying the **FTJ** behavior.

While the 2T1C cell is certainly interesting as a characterization target and versatile circuit, its use in commercial devices is less clear, due to its decreased density compared to 1T1C cells, at least for the **DRAM** operation mode (other modes have the advantage of direct access to both poles of the capacitor for programming). This can be compensated by designs exploiting the computing potential of the 2TnC cell. It also remains to be seen how capacitor scaling to smaller sizes affects reliability, especially as it increases device-to-device variability[Den+20].

Cell density could be compensated by increasing the number of levels to some extent: larger capacitors induce larger readout currents, which are easier to discriminate. This also comes with longer programming times, which can also help with multi-level cells, but might require higher voltages to reach acceptable performance. The optimal choice is probably application-specific, notably depending on technological node used for co-integrated logic: control circuits of higher complexity may be able to account for increased variability. A **DSE** approach would likely be appropriate for determining the trade-offs between capacitor size and the number of levels.

It is also important to stress that the density lost due to the added transistor is small: the access transistor Q_A is relatively small compared to the readout transistor, and the design decouples transistor sizes from capacitor sizes to an extent, compared to **FeFETs**, as detailed in chapter 4. Hybrid circuits closer to the 2T1C concept can also be envisioned, sharing a readout transistor between multiple 1T1C cells.

Chapter 4

Ferroelectric transistor-based designs

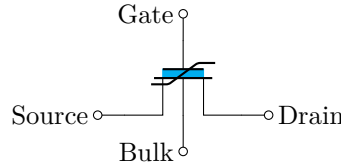
Contents

4.1 Introduction to FeFET circuits	79
4.1.1 Programming the ferroelectric oxide	80
4.1.2 V_{th} shifting	80
4.1.3 Comparison with CMOS-based logic	82
4.2 1T-FeFET memory	83
4.2.1 Operating principle	84
4.2.2 Comparison with other floating-gate transistor memories	85
4.2.3 Possible hybrid operation mode	85
4.3 Transresistance circuits	86
4.3.1 Complementary logic with p-FeFET	86
4.3.2 Resistive logic	87
4.3.3 Dynamic logic	87
4.3.4 Pass-transistor logic	89
4.4 Non-volatile FeFET-based logic gates	89
4.4.1 NV-NAND2	89
4.4.2 NV-AND2	90
4.4.3 NV-XOR2	90
4.5 FeFETs as add-on technology	91
4.5.1 Black & Das memory cell as a checkpointing mechanism	91
4.6 Convolutional Image Filter with FeFET-based Logic-in-Memory	92
4.6.1 Choice of a convolutional image filter	92
4.6.2 Filter architecture	95
4.6.3 FeFET-based Logic-in-Memory multiplier design	98
4.6.4 Validation in simulation and identified issues	103
4.6.5 Results	106
4.7 Conclusion	108
4.7.1 FeFET-based logic	108
4.7.2 Image filter	110
4.7.3 FeFET-based memories	110

4.1 Introduction to FeFET circuits

In this chapter, the benefits of using FeFETs as single-transistor (1T) memories for normally-off and LiM computing are investigated. This is achieved by leveraging the natural advantage of FeFETs, i.e. the direct combination, at the device level, of logic switching (transistor) and non-volatile information storage (ferroelectric layer).

As a preamble, this chapter begins with a discussion of the main tenets of FeFET circuit operation, in particular how to program the ferroelectric oxide when it is embedded in the transistor gate stack, and the principles surrounding the desired physical effect, i.e. the shifting of the transistor threshold voltage through the state of polarization of the ferroelectric oxide.



CIRCUIT 4.1: FeFET symbol and connections, as presented in section 2.4

4.1.1 Programming the ferroelectric oxide

To polarize¹ the ferroelectric oxide, an electric field must be applied across it. A logic high state “1” is stored when applying a high voltage pulse ($V > V_C$, so that $E > E_C$) to the gate of the transistor. A logic low state “0” is obtained by applying reversing the direction of the electric field being applied.

This can be challenging to perform, as providing negative voltages requires more complex power supply and management circuits. An alternative approach is to apply the ground potential to the gate, and a positive voltage to the bulk, source and drain.

While applying a voltage to any one of these three terminals should enable the ferroelectric oxide to be programmed in most cases, the electric field is more homogeneous when applying it to all three. As detailed in subsection 2.4.2, this is particularly true in the case of a gate stack that does not contain a metal layer between the ferroelectric oxide and gate oxide, as that metal electrode forces the electrical field to be applied vertically across the ferroelectric oxide.

In this chapter, the polarity of programming pulses is defined as being the same as that of the gate potential, when taking the opposite side of the transistor gate stack as the voltage reference. For instance, when programming the ferroelectric layer by applying a voltage between the bulk of the transistor (V_{BLK}) and the gate (V_G), the following is obtained:

$$\begin{cases} V_G - V_{BLK} > V_C^+, & \text{Positive pulse} \\ V_G - V_{BLK} < V_C^-, & \text{Negative pulse} \end{cases} \quad (4.1)$$

$$(4.2)$$

4.1.2 V_{th} shifting

The operating principle behind most FeFET circuits is V_{th} shifting: depending on the polarization state of the ferroelectric layer, the transistor threshold voltage can be increased or decreased.

N-channel FETs are considered non-conducting when the applied gate voltage is lower than their threshold voltage, which depends on oxide thickness and gate length, as well as technological parameters.

Re-polarizing the ferroelectric layer releases charges in the transistor gate, which is a floating node. These charges in turn help to establish conduction in the channel, requiring a lower input voltage to reach the threshold voltage on the floating transistor gate if the previously applied polarization pulse was positive, or a higher one if it was negative.

V_{th} modulation can affect the behavior of transistors in a non-volatile manner. To create FeFET-based circuits, it is necessary to choose both the starting threshold voltage, and the shifting amount.

Analog V_{th} shift control

If the field applied across the ferroelectric gate oxide can be controlled finely enough, intermediate polarization states are achievable, allowing analog control of the new, shifted V_{th} value. The range of possible polarization states increases as the number of ferroelectric domains increases. Moreover, a larger V_{th} shift is achievable if the charges released by the ferroelectric oxide have a greater impact on the transistor gate. This is also achieved by designing, the ferroelectric capacitance larger than the MOS capacitance, as discussed in section 3.3.2.

¹The terms “programming”, “writing” and “erasing” are used in this section relatively interchangeably, according to the rules from section 2.1.1.

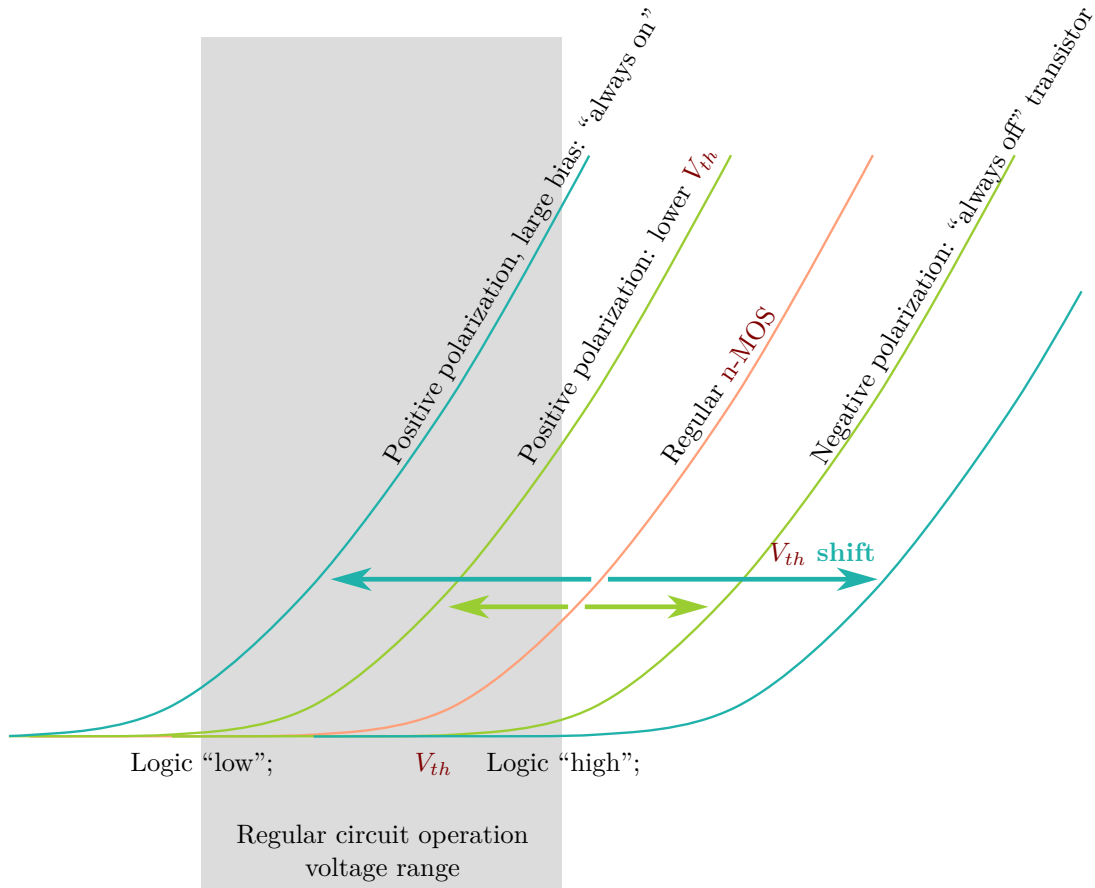


FIGURE 4.1: Illustration of V_{th} shifting: a regular **n-MOS**'s I_{DS} — V_{GS} characteristic (●) is shown next to the shift that would be provided by two different ferroelectric layers, capable of releasing a relatively small (●) and large (●) amount of charge, respectively. In practice, the transistor's **threshold voltage** will likely be chosen higher than that represented above, so as to provide improved switching characteristics after programming the ferroelectric oxide.

By choosing the initial **threshold voltage** and adjusting the capacitance ratio of the transistor and ferroelectric capacitor, the resulting **FeFET** can switch between two of the three possible operation modes visible above: regular **n-MOS** operation, always-on or always-off **n-MOS**.

Intermediate polarization states can be reached by controlling both pulse height (voltage) and length (duration). As ferroelectric domains are sensitive to electric field intensity, pulse voltage have a direct impact on their polarization. However, relatively large capacitances from the ferroelectric oxide and MOS transistor gate limit the rate of voltage increase, making them sensitive to pulse duration as well. As repolarization is not instantaneous either, this may be a contributing factor to time-sensitivity.

Lastly, it is possible to repolarize a ferroelectric by applying a pulse train of similar voltages (accumulative switching), though the Preisach model is unable to model this behavior [Den+20].

Analog control may be interesting for neuromorphic computing[Yoo+19], as well as multi-level memories (MLC).

Dual-state V_{th} shift control

In other cases, where the polarization can only be controlled using two programming voltages corresponding to two ferroelectric states (high and low), the desired positions of the new threshold voltages need to be chosen. As illustrated in Figure 4.1, the possibilities for the shifted threshold voltage V'_{th} are the following:

- $V'_{th} < L < V_{th} < H$ Always on
- $L < V'_{th} < V_{th} < H$ Reduced V_{th} (normal MOS operation)
- $L < V_{th} < V'_{th} < H$ Increased V_{th} (normal MOS operation)
- $L < V_{th} < H < V'_{th}$ Always off

In the list above, H and L stand for logic high and logic low voltage values, respectively. V_{th} represents the threshold voltage of a regular transistor. In an ideal case, the FeFET's non-shifted threshold voltage (V_{th}^F) should be chosen independently of it, and set to the midpoint of both desired shifted threshold voltages $V_{th}^{FL} < V_{th}^{FH}$. The actual threshold voltage shift depends on the coercive voltage used for programming, and can be used in an analog, or multi-level way[Yoo+19]; though this use-case will not be detailed.

There are therefore three interesting cases to consider:

- Always on / Regular MOS: $V_{th}^{FL} < L$ and $L < V_{th}^{FH} < H$
- Regular MOS / Always off: $L < V_{th}^{FL} < H$ and $H < V_{th}^{FH}$ and
- Always on / Always off: $V_{th}^{FL} < L$ and $H < V_{th}^{FH}$, which corresponds to the largest V_{th} shift

This chapter focuses on the second use-case, which can be seen as a toggle-able n-MOS transistor. This choice was made due to it being the kind of device available for fabrication. Different circuits are achievable in the other cases. The last case of an always on or always off n-MOS transistor could for instance be used in switch boxes for FPGA interconnect circuits, particularly as the $R_{DS,on}/R_{DS,off}$ ratio may be engineered to be relatively large in that case. Achievable configurations depend on technological and design-time parameters (V_{th}^F and maximum V_{th} shift), as well as the voltage applied at programming time. Therefore, all three configurations (always on, always off, regular transistor) may be accessible to the same device.

4.1.3 Comparison with CMOS-based logic

Advantages compared to CMOS-based logic

The ability to store information by shifting the transistor's threshold voltage reduces the need for supplying logic operands from memory. Being non-volatile, this also reduces the need for dedicated Non-Volatile Memory, as this functionality is then built into the circuit. This has the potential to:

1. simplify board design, as Non-Volatile Memories are often separate chips

2. reduce power consumption, as external memories do not need to be powered and read from
3. increase performance, as the processing unit does not have to wait for memory accesses, also further reducing power consumption

As will be detailed, FeFET-based circuits also have the potential to use fewer transistors than their CMOS counterparts: for instance, a TCAM design[Yin+19] can be reduced from 16 MOSFETs per bit, to 2 FeFETs per bit, plus three MOSFET.

Deficiencies compared to CMOS-based logic

Complementary FeFET logic was not an option at the time of this work, as p-FeFET were unavailable, as detailed below. This severely limited the static power efficiency achievable, as will be detailed in the next section. The minimal capacitor footprint required due to domain size is also relatively important (on the order of 200 nm of diameter as seen in section 2.1.1), which produces large transistors when capacitance-matched. The larger sizes of these transistors, as compared to transistors at the minimal dimensions allowed by the technology, are therefore slower and require more energy during logic switching. Finally, programming the ferroelectric oxide requires relatively large voltages (of about 3 V to 4 V), that increase with the thickness of the gate stack, as the Coercive Electric field of the ferroelectric material needs to be reached. This requires compatible circuitry, both to generate and to withstand such voltages, generally incompatible with advanced technology nodes.

Technology process and p-FeFET availability

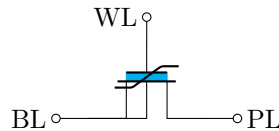
All the work carried out in this chapter is based on the GlobalFoundries 28SLP technology standard design kit with the addition of a FeFET module[Bey+20]. This includes design, transistor-level simulation, and fabrication. The ferroelectric layer is simulated using the Preisach approach, as described in subsection 2.2.2.

With this technology process, transistor V_{th} is around 1 V, with a ferroelectric Coercive Electric field $E_C \approx 1.2 \text{ MV cm}^{-1}$, yielding FeFET gate/bulk programming voltages of about $\pm 3 \text{ V}$.

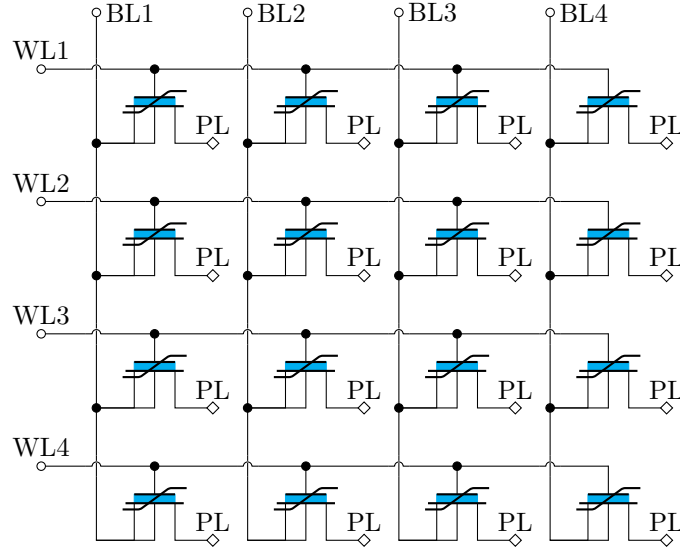
During the first attempts at integrating ferroelectric HfZrO₂ on the gate stack, the focus was on the realization of n-FeFET devices for utilization in memory arrays[Mül+21] such as those described in section 4.2. Although progress has been made recently[Kle+21] towards fabricating fully integrated p-FeFETs manufactured in the same technology, with characteristics similar to n-FeFETs, they were not an available option at the time of design. This oriented the following designs towards alternative architectures such as resistive or dynamic logic, as detailed in section 4.3.

4.2 1T-FeFET memory

One of the simplest FeFET-based circuits is the 1T-FeFET memory, which bitcell consists of a single FeFET. Its operating principle is very similar to that of flash memory, as will be detailed in subsection 4.2.2.



CIRCUIT 4.2: 1T-FeFET bitcell showing WL, BL and PL connections. Note that the bulk is connected to BL in order to facilitate programming.



CIRCUIT 4.3: 4×4 memory array of bitcells from Circuit 4.2.

4.2.1 Operating principle

Circuit 4.2 shows the base bitcell, while Circuit 4.3 shows how that bitcell can be integrated in an array. Other programming and connection schemes are possible[Rei+19], but do not substantially differ from the one presented in these schematics.

Read operation

Assuming fabrication parameters to have been chosen such that FeFET in the memory array function either as regular n-MOS, or to be always-off, their conduction state occurs when the following conditions are both true:

- The ferroelectric is polarized in the “high” state
- The WL signal is a logic high.

Therefore, similarly to a 1T1C array, individual bitcells can be addressed at the intersection of an active WL and BL: a whole WL is addressed, making all logic-high FeFETs on that WL conduct. In this specific implementation, a positive voltage is applied PL, while BL, which is connected to the bulk, is pulled low.

The transistor at the intersecting BL and WL conducts if a logic high state was stored, allowing the detection of a current between the active BL and the PL.

If unselected BLs are pulled low, other transistors from the same WL may conduct. While this is not an issue for detecting the state of the selected bitcell, it will cause greater power consumption: these lanes can be pulled high or left floating instead.

For pursuing greater power efficiency, a voltage readout can be performed instead of a current one. This requires connecting the bulk separately to allow higher output voltages. Moreover, the reading scheme described above assumes PL capacitance is higher than BL: Achieving optimal reading speed may require applying a voltage to BL and reading the output on PL, depending on respective BL and PL capacitance.

Write operation

To perform a write operation, a field needs to be applied across the ferroelectric oxide. That is, from the WL contact to the source, drain or bulk of the FeFET, with a preference for the bulk. This is the reason for which it was chosen to connect the bulk to the BL in Circuit 4.2.

PL is common to all devices in the array and for this reason should be left floating, while V_C is applied between the selected BL and WL. To program a “logic high”, WL is pulled up

to V_C , while BL is pulled down to zero. The opposite is performed to store a “logic low”, avoiding the use of negative voltages.

As only one of the write operations require changing the bulk (if separate) and PL potential from the one used for reading, operations programming “logic low” values in nearby cells should be batched together for better power efficiency and latency, similarly to a “block erase” operation on $flash$ memory. Likewise, pre-setting unused memory areas to a known value, similar to “TRIM” operations in $flash$ memory may also improve these metrics.

4.2.2 Comparison with other floating-gate transistor memories

The main difference with other $FGMOS$ memory technologies (including $flash$, UVPROM, EEPROM) lies in the mechanism used to change the quantity of electric charge on the floating transistor gate: the aforementioned memories usually use Fowler-Nordheim tunneling or hot-carrier injection to progressively store or empty charges to and from the floating gate. $FeFETs$ use polarization reversal as the mechanism, exchanging charges from the ferroelectric surface with the floating metal gate.

Advantages include lower programming voltages, and $CMOS$ compatibility in the case of $HfZrO_2$, possibly associated with higher endurance, increased speed and reduced power consumption.

Drawbacks include the inability to compensate for charge build-up or loss in the floating gate: whereas other technologies can empty the floating gate completely, $FeFET$ -based memories cannot control the absolute quantity of charges, either to get rid of an excess of charges or to compensate for a lack of charges. This can lead to unwanted V_{th} shifting, and endanger the gate oxides in case of excessive charge build-up. Regularly repolarizing the ferroelectric oxide (for instance, by changing the bit mapping) may lower this effect.

Flash-like $NAND$ and NOR arrays are also possible, provided the V_{th} shift is small enough to allow the transistor to be set to a conductive state without erasing the ferroelectric polarization.

4.2.3 Possible hybrid operation mode

In light of the issues mentioned above, hybrid approaches can also be explored to improve performance metrics. It might be possible to use Fowler-Nordheim or ferroelectric junction tunneling to pre-charge the floating gate to a desired state, but another possibility is the addition of an access transistor to the floating transistor gate, as in the case with the 2T1C circuit from [section 3.5](#).

While this is not possible with traditional $flash$ memory or other $FGMOS$ -based memories as the additional leakage current would severely impact retention, ferroelectric memories also store the state inside the ferroelectric oxide polarization, which is not endangered by leakage currents. Therefore, even if the normal 1T- $FeFET$ read operation mode cannot succeed as charges on the floating gate have dissipated, the previously stored value can be measured by operating the ferroelectric oxide in a 1T1C mode, performing a destructive read operation on the ferroelectric oxide as described in [section 3.5.1](#) and [section 3.2](#).

Essentially, this would allow the ferroelectric memory to operate in a fast, $DRAM$ -like fashion, with the possibility to store data in the volatile paraelectric, or the non-volatile ferroelectric part of the $FeCap$ array by controlling the writing voltage, using the transistor as a non-destructive reading mechanism for the floating gate node, as in the case of 1T- $DRAM$ [Giu21; LFZ11] (Floating Body 1T- $DRAM$).

Performance figures for such a hybrid memory remain to be determined, including maximum retention time for $DRAM$ -like operation, operation speed and energy consumption. Retention time dictates the frequency at which charges on the floating gate should be refreshed, it is therefore the most crucial parameter. Depending on the use-case, refresh could be handled:

- in a $DRAM$ -like fashion, by compensating the charge leakage at the floating node;
- by performing a ferroelectric 1T1C-like destructive read of the ferroelectric;
- by letting the charges completely evaporate, and only performing 1T1C-like destructive reads, with possible refresh afterwards.

The most suitable refresh mode can be chosen depending on the workload, and multiple combinations could be used on a single hybrid memory array. **DRAM**-like operation is best suited for memory areas that either necessitate low-latency access or are read and written frequently, such as a computer program stack, and could forego non-volatility by using lower programming voltages, not reprogramming the ferroelectric oxide in exchange for less energy expenditure and increased throughput, only “**checkpointing**” specific states: the values stored inside the ferroelectric oxide and the floating gate can be different, allowing two bits to be stored per bitcell, one of which in a non-volatile way inside the ferroelectric polarization.

Destructive refresh can be useful for “**pre-warming**” a memory area that is expected to become latency-sensitive, while 1T1C operation mode can be reserved for archived data, where latency or endurance are not critical. As previously mentioned, the floating gate of such memory areas could be re-purposed as **DRAM** by operating under V_C , leaving the persistent area unmodified, but increasing the amount of volatile memory available on the system.

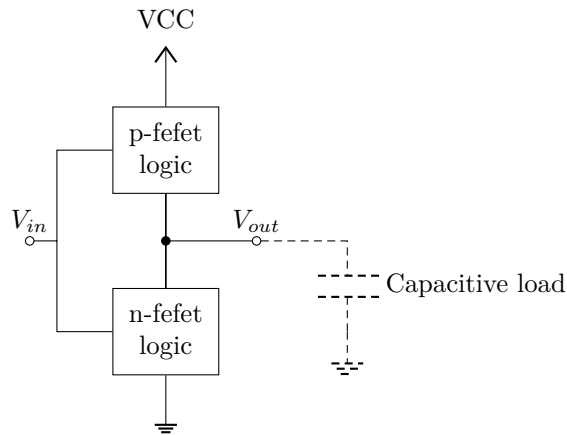
4.3 Transresistance circuits

FETs, including **MOSFETs** and **FeFETs**, are transconducting devices: their current output depends on the input voltage (transconductance). To cascade multiple **FETs**-based circuits, the output current needs to be converted to an input voltage for the next circuit (transresistance).

This section provides a brief overview of available strategies for outputting computation results as a logic voltage.

4.3.1 Complementary logic with p-**FeFET**

While p-**FeFET** were not available as design options during the course of this thesis, as explained in section 4.1.3, they are theoretically perfectly achievable, and have been recently demonstrated[Kle+21]. This opens the door to **CMOS**-like c-**FeFET** logic, as represented in Circuit 4.4, using both n-**FeFET** and p-**FeFET** to lower static power consumption.



CIRCUIT 4.4: Transresistance circuit with complementary **FeFET**. The necessary capacitive load is represented dashed.

For complementary logic to function properly, p-**FeFET** need to be designed as complementary devices to n-**FeFET**. Table 4.1 summarizes the necessary logic states. As visible in the table, both circuits need to be activated after a logic high programming pulse is received.

As described in subsection 4.1.2, applying such a pulse increases the actual gate voltage, lowering the threshold voltage of the **FeFET**. Increasing the threshold voltage of a p-**FeFET** means that it will remain in a conductive state regardless of the input value. Therefore, the same approach is compatible with both n-**FeFET** and p-**FeFET**: with the same initial V_{th} , the addition of a ferroelectric layer will disable the ferroelectric transistor after a negative programming pulse. p-**FeFET** will remain in a conductive state, while n-**FeFET** will remain in a blocked state.

B	A	resistance state
0	0	High
0	1	
1	0	
1	1	Low

(a) n-FeFET

B	A	resistance state
0	0	Low
0	1	
1	0	
1	1	High

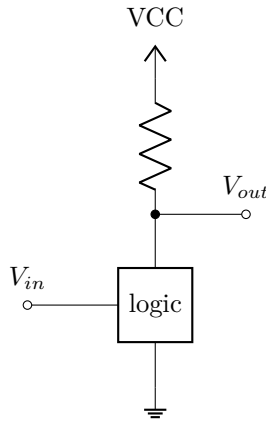
(b) p-FeFET

TABLE 4.1: n-FeFET (4.1a) and p-FeFET (4.1b) with complementary behavior: normal CMOS behavior is observed when the ferroelectric oxide is in the “high” logic state (meaning the FeFET received a high-voltage positive pulse on the gate). B, A, resistance state are respectively the states of the ferroelectric oxide, the signal received on the FeFET gate, and the corresponding channel conductivity.

The compatibility of these devices owes to their normal operation range in CMOS circuits; in fact the ferroelectric voltage-enhancing behavior can be shared across multiple transistors, allowing a single ferroelectric capacitor to be shared across both p- and n-PsFeFET, including during the programming phase, potentially drastically simplifying peripheral and control circuitry. This is detailed in subsection 3.3.4, though the same retention considerations apply.

4.3.2 Resistive logic

A simple pull-up resistor can be used to output a logic high by default. When the logic circuit is conducting, the output is pulled low. This is the simplest way to obtain a voltage output from conductivity changes, but comes at the cost of relatively high leakage currents. This corresponds to the RTL (Resistor-Transistor Logic) class of digital circuits.



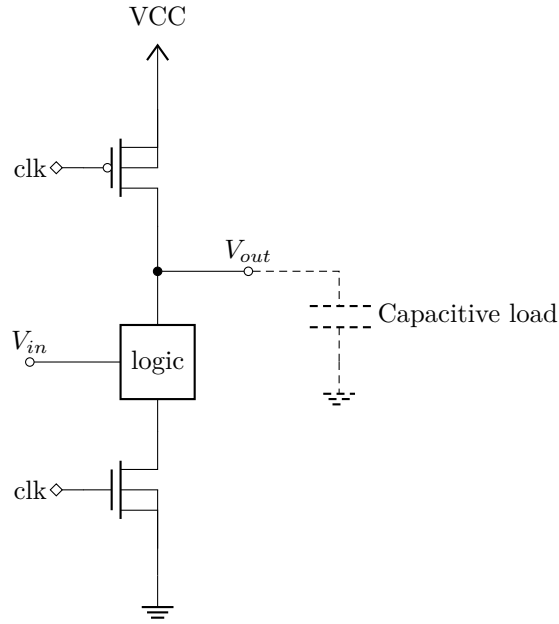
CIRCUIT 4.5: FeFET circuit with pull-up resistor for handling transresistance.

4.3.3 Dynamic logic

In the absence of p-FeFET, static power consumption can be reduced by disabling the power supply, relying on the input capacitance of following transistor gates to hold computed output values.

As presented in Circuit 4.6, the clock signal first allows the input capacitance of the next stage to be charged (pre-charge phase). Then, depending on the conductance value of the logic network, the output can be pulled back to a low value again (evaluation phase).

Dynamic logic is promising for energy efficiency as it severely restricts static power consumption, but it introduces synchronization issues for cascading logic gates, as the clock signal must propagate at the same rate as the output signal. Synchronizing input signals when cascading logic gates with varying propagation delays can also lead to choosing the clock period based on the worst-case propagation delay, which reduces throughput.



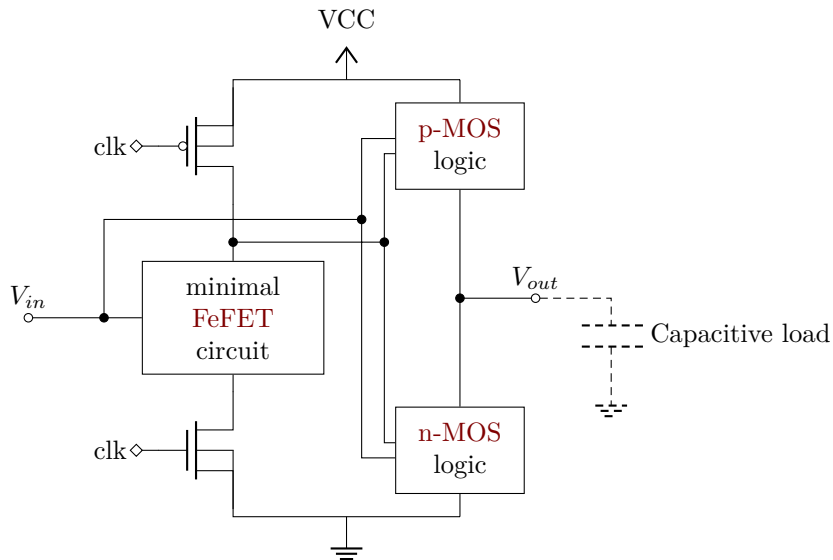
CIRCUIT 4.6: Transresistance with dynamic logic architecture

Hybrid dynamic logic with CMOS stages

Cascading multiple dynamic logic blocks is possible through the use of synchronization stages. This can be achieved by pipelining the computation path, adding latches to store operands and synchronize inputs.

However, synchronization stages increase the latency, and are not always justified when used for synchronizing small delays, on the order of a fraction of a clock signal.

Instead, a hybrid approach illustrated in **Circuit 4.7** can be taken, reducing the **FeFET** count to the minimum and using **CMOS** where the comparative advantages of **FeFET** are not leveraged. This is particularly useful when attempting to create asynchronous sequential logic.



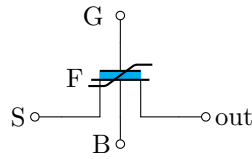
CIRCUIT 4.7: Hybrid CMOS and dynamic logic architecture

4.3.4 Pass-transistor logic

FeFET-based pass-transistors

FeFET can be used both for controlling pass-gates, and directly as pass transistors. As illustrated in **Circuit 4.8**, pass-transistor logic implements a completely different approach to transmitting the signal. That approach sidesteps the need for complementary logic, and often reduces the transistor count compared to other static power reduction techniques. Since pass-transistor logic does not rely on periodically charging and discharging a node, it is not subject to dependency issues.

This comes with the usual trade-offs pertaining to pass-transistor logic: lower **fan-out**, and better transmission of low logic levels than high logic levels when p-**FET** are not used. These issues usually require mitigation techniques such as level restorers or output buffering after a low number of cascaded gates.



(a) Schematic

B	F	G	S	out
0	1	1	0	0
		1	1	h
	0	0	-	Z
1	-	-	-	-

(b) Truth table

CIRCUIT 4.8: Pass transistor logic circuit, with **FeFET** as a pass-transistor, and associated logic table. In the truth table, “-” represents unimportant states, “Z” is high-impedance, and “h” is weak logic high.

Circuit 4.8 conducts when both the gate and ferroelectric are logic high: $\text{out} = G \cdot F$. For the sake of completeness, bulk and source terminal potentials are included in the logic table, as these signals can be combined to create more logic functions. This circuit would typically be combined with others that override it during its high-impedance “Z” state.

4.4 Non-volatile **FeFET**-based logic gates

The V_{th} shift principle detailed in **subsection 4.1.2** can be exploited to perform a logic operation between the **FeFET** input value and that stored in the ferroelectric polarization[**Mar+21; Bre+18; OCo+18**]. By convention, a logic stored value “1” (or high) corresponds to the value programmed by the application of a voltage-boosted logic high signal on the gate input, and a stored “0” corresponds to the application of the opposite potential difference. In other terms, when programming a **FeFET** with positive and negative voltages, storing “1” is performed by applying a positive V_C on the **FeFET** gate, and storing “0” is performed by applying a negative V_C .

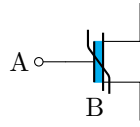
This section describes the pull-down networks for multiple logic gates: a high conductivity state will translate into a logic output low. A transresistance circuit or the dual p-**FeFET**-based circuit can be employed in the pull-up branch, as discussed in **section 4.3** and **subsection 4.3.1** respectively. This is presented separately, as p-**FeFET** were not commercially available in the 28SLP technology when designing these circuits.

Accordingly, the following truth tables present both the **Resistance State (RS)**: high (H) or low (L) resistance of the pull-down stage, and the **Voltage Output (VO)** after transconducting it with the pull-up stage.

4.4.1 NV-NAND2

The simplest **FeFET**-based logic gate is also a universal gate: the **NAND** gate can be constructed from a single ferroelectric transistor, as shown in **Circuit 4.9**.

The **FeFET** conducts only if the V_{th} was shifted in the regular transistor operation zone, and the input of the transistor is a logic high. Therefore, the circuit pulls the output low only in that case, and a logic **NAND** is performed, as the output stays pulled high in every other case.



(a) NV-NAND2 Circuit

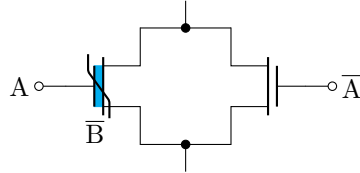
A	B	RS	VO
1	1	L	0
0	1	H	1
1	0	H	1
0	0	H	1

(b) NV-NAND2 Truth table

CIRCUIT 4.9: **NAND FeFET**-based pull-down circuit. “B” is the non-volatile operand stored ahead of time in the **FeFET**.

4.4.2 NV-AND2

The right transistor in **Circuit 4.10** conducts if input A is low, excluding all such cases. The left **FeFET** conducts if B is low (stored \bar{B} high), in the case where A is high. This creates an AND logic gate, as the only non-conducting case (for which the output will remain pulled high) is when both inputs are high.



(a) NV-AND2 Circuit

A	B	Left	Right	RS	VO
1	1	H	H	H	1
0	1	H	L	L	0
1	0	L	H	L	0
0	0	H	L	L	0

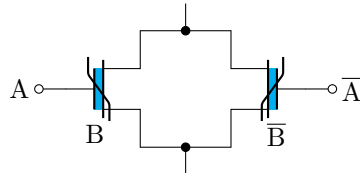
(b) NV-AND2 Truth table, including **resistance state** of left and right branches

CIRCUIT 4.10: AND **FeFET**-based pull-down circuit. Note the complementary value \bar{B} stored in the left ferroelectric transistor, and the complemented input \bar{A} of the right branch. The output depends on the **resistance state** of both branches: if either is low, the total equivalent resistance **RS** will be low.

Depending on the circuit controlling the inputs and voltage shifting, this circuit can be more or less practical to realize. Of note is the inverted value stored in the **FeFET**. If a voltage shifter is dedicated to the circuit, or if the programming phase accepts different inputs than during regular circuit operation, storing a complemented value should not be an obstacle. Inputting the complementary value on the right transistor however can also require an additional inverter, adding to the overall transistor count.

4.4.3 NV-XOR2

The **XOR** circuit presented in **Circuit 4.11** is similar to the AND logic gate presented before, with the addition of a second ferroelectric transistor in the parallel branch. This second circuit has both inputs (the **FeFET** input and stored value) inverted, and therefore conducts in the opposite case where both inputs are logic low. The pull-down network conducts when either branch is in the low **resistance state**, as illustrated in **Circuit 4.11b**.



(a) NV-XOR2 Circuit

A	B	Left	Right	RS	VO
1	1	L	H	L	0
0	1	H	H	H	1
1	0	H	H	H	1
0	0	H	L	L	0

(b) NV-XOR2 Truth table, including **RS** of left and right branches

CIRCUIT 4.11: **XOR FeFET**-based pull-down circuit. The operation being performed is $\overline{A \cdot B + \bar{A} \cdot \bar{B}}$, analogous to **XOR**: $A \cdot \bar{B} + \bar{A} \cdot B$.

A logical **XOR** can thus be constructed with two **FeFET**, plus one inverter. As for other **FeFET**-based circuits, additional multiplexers or level shifters are likely necessary, and the inverter may need to output higher voltages during programming.

4.5 FeFETs as add-on technology

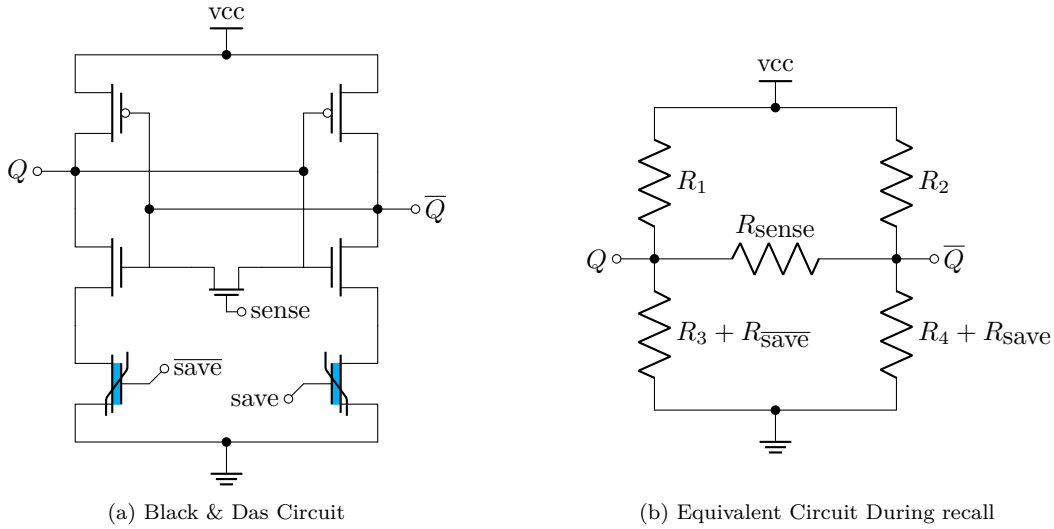
Thanks to the simplicity of **FeFET**-based memories, that consist of a single ferroelectric transistor, they can be used to enhance classical memory architectures in order to provide additional functionalities, such as non-volatility.

4.5.1 Black & Das memory cell as a **checkpointing** mechanism

As an illustration of this, the “Black & Das”[BD00]-like structure as presented in [OCo+18] is a modified **SRAM** cell that stores an additional bit of data in a **FeFET**-based power supply network, as visible in **Circuit 4.12a**.

This cell works by unbalancing the symmetry of traditional **SRAM** cells: both **n-MOS** transistors are in series with a **FeFET** that store a different value, leading to different equivalent resistances, as shown in **Circuit 4.12b**. This affects the drive-down ability of the cross-coupled inverters, as shown in **Figure 4.2b**. The side with the lower resistance in the pull-down network will therefore be biased towards that state on start-up, when $Q = \bar{Q}$. For instance, with the left branch in a low **resistance state** (corresponding to a positive programming voltage applied on the left, $\overline{\text{save}}$ terminal), output Q will be pulled low, and \bar{Q} high.

This “startup recall” can be manually triggered without cutting the power supply (and waiting for both Q and \bar{Q} to discharge) by short-circuiting both outputs with an additional “sensing” transistor, visible in the middle of **Circuit 4.12a** and represented by R_{sense} in **Circuit 4.12b**. Triggering this transistor forces the circuit into the metastable state, and the imbalance in driving **n-MOS** transistors branches will steer the output towards the stored state once the sense transistor is turned off again, as plotted in **Figure 4.2a**.



CIRCUIT 4.12: Black & Das **FeFET**-enhanced **SRAM**, and equivalent resistive circuit during sensing.

Some optimizations can be performed on the circuit presented in **Circuit 4.12a**, leading to footprint and speed improvements: the transistor can be foregone if the recall operation is only expected to happen on start-up, when enough time has passed to allow both outputs to discharge. A single **FeFET** can also be kept, provided $R_3 + R_{\text{FeFET DS}(0)} > R_4 > R_4 + R_{\text{FeFET DS}(1)}$, with the same nomenclature as in **Circuit 4.12b**, and $R_{\text{FeFET DS}(0)}^{\text{HIGH}}$ the equivalent resistance of the **FeFET** after programming it with a HIGH_{LOW} gate voltage. Circuit performance can be further improved if parameters are chosen to keep a low equivalent resistance, as higher resistance in the pull-down area will slow down **SRAM** operation. This

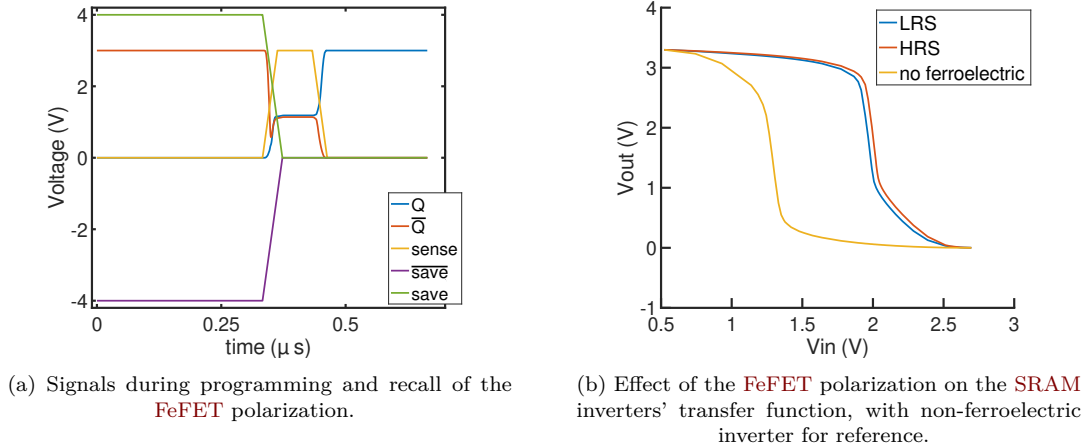


FIGURE 4.2: Black & Das cell behavior: A positive programming signal switches the FeFET into a *Low resistance state*, a negative one into a *High resistance state*. The programmed state can be recalled at any point of the SRAM operation by short-circuiting the cell using the *sense* signal. These simulations were performed with a Landau model.

can be achieved by adjusting the FeFET gate, doping it more, making it shorter or wider, or constantly supplying a positive voltage on the gate terminal during operation. Current leakage through the FeFET in the off-state is not a concern here, as long as the above condition is respected: better performance may be achieved by using a highly-doped, short channel FeFET.

A use-case for that kind of architecture is to store default values in SRAM memory, such as a bootloader. It can also be useful for *normally-off* computing: in the event of a power cut, values in the SRAM are erased, but not those stored in the FeFET. This can provide a *checkpointing* mechanism in case of power loss, or if intermediate operations need to be undone and the SRAM restored to a previous state. As a hybrid architecture, SRAM provide their unrivaled speed and endurance, while FeFET provide non-volatility. The use of PsFeFET may also help reduce the additional footprint of the FeFET by placing ferroelectric capacitors above the SRAM transistors.

4.6 Convolutional Image Filter with FeFET-based Logic-in-Memory

As a practical application of FeFET-based logic gates, it was decided to conduct work towards a technology demonstrator within the 3εFERRO project.

To demonstrate the FeFETs performance potential for application in LiM designs, the realization of a real-time image filter circuit demonstrator was targeted. In this application, the image filter can be reconfigured to hold e.g. edge detection as well as sharpening or blurring filters, depending on the use-case.

The effort being led by NaMLab, ECL-INL contributed select logic blocs, and handled final design validation, identifying multiple critical issues that would have prevented correct operation of the device.

4.6.1 Choice of a convolutional image filter

After investigating multiple options for demonstrating a credible use-case for FeFET-based circuits in a working prototype, the choice converged towards a convolutional image filter. FeFET-based circuits take two operands: a “static” one stored in *Non-Volatile Memory* within the oxide polarization and the floating gate, as well as a “dynamic” one that is provided as an input to the FeFET gate.

This directs the choice of circuit towards signal processing, as the set of operations is generally determined ahead of time and remains static during the duration of the processing, while the signal is streamed from an input (often from a sensor), processed, and output. The non-volatility of the FeFET makes **normally-off** appealing if it can be leveraged to store processing parameters, as they do not need to be read from an external memory during the start-up sequence.

Digital Signal Processor (DSP) are commonly used in the industry to perform signal processing operations. Among the possible algorithms, **Finite Impulse Response (FIR)** filters are one of the most common, and only require delaying the input signal, multiplying samples with fixed coefficients, and summing the output. They are versatile enough to be useful in a wide variety of contexts, from audio processing (echo cancellation, noise filtering, equalizing), to control applications (PID control), and image processing.

The specific choice was made to implement a convolutional filter suitable for image processing, as convolution is a memory-intensive operation. A simple change of coefficients can generate a variety of strikingly different outputs, as illustrated in **Figure 4.3**. As such, they are commonly in image processing, as well as machine learning with **Convolutional Neural Networks (CNNs)**. By setting the topology of the filter, design complexity is reduced, while demonstrating the implementation of a non-trivial algorithm. The filter can also be re-purposed for other tasks than image processing, as long as the input data and coefficients can be adjusted to the new application: for instance, it is directly usable as a single-layer perceptron, which hints at further potential uses in machine learning and neural network applications.

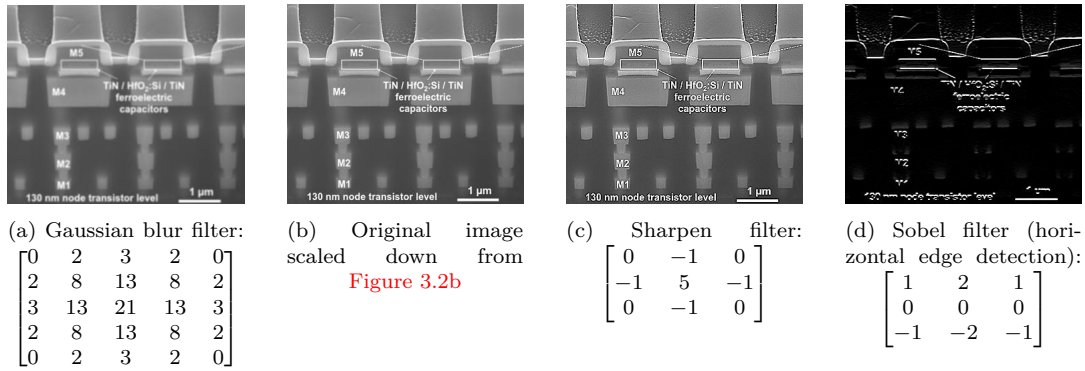


FIGURE 4.3: Examples of image filtering operations and corresponding **kernels**, generated with code from **Listing A.10**.

Convolution operation in one dimension

Like with other **FIR** filters, the discrete convolution operation operates on a series of input samples by multiplying them with coefficients and summing them. It corresponds to the simplest **FIR** filter category, the direct form, where no further multiplication operation is performed on intermediate results.

The continuous form of convolution between two time-dependent signals $f(t)$ and $g(t)$ can be written as:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) \cdot g(t - \tau) d\tau$$

In the time-discrete domain, f_{k+n} representing the $(k+n)^{\text{th}}$ sample:

$$(f * g)_k = \sum_{n=-\infty}^{\infty} f_n \cdot g_{k-n} = \sum_{n=-\infty}^{\infty} f_{k-n} \cdot g_n$$

In practice, only the input signal f is infinite as it is often a stream of data output by a sensor. g is the filtering signal and of finite length: $g_k = 0$ for $k > l$ with l the length of the signal.

$$(f * g)_k = \sum_{n=1}^l f_{k-n} \cdot g_n = f_{k-1} \cdot g_1 + \dots + f_{k-l} \cdot g_l$$

l samples from the input f signal therefore need to be stored at most (as it is common to pad signals with zeroes), with l multiplications and $l - 1$ additions.

Convolution of a two-dimensional image

The process for performing a convolution operation on two images is similar, as illustrated in **Figure 4.4**. As the goal of such a filtering technique is to apply a transformation uniformly to an input image, the applied transformation is often a much smaller image called a **kernel**.

The same computation as the 1D convolution is performed independently on multiple image lines with the corresponding filter **kernel** line, and the results are summed to give a new pixel value that depends on every neighboring pixel value. There are, however, a few differences that must be pointed out, arising from how images are stored in memory and transmitted. Traditionally, and owing to cathode ray tube heritage, individual pixel (brightness) values are sent line-by-line from the top-left to the bottom-right of an image. For an image of W pixels wide, there are therefore W pixels transmitted between a given pixel and the one directly below it. A 2D image filter therefore corresponds to a 1D convolutional filter with zero-padding of these W intermediate pixels. While there is no need to perform multiplications by zero within these padded areas, the corresponding pixel values still need to be either stored temporarily, or re-transmitted at a later time.

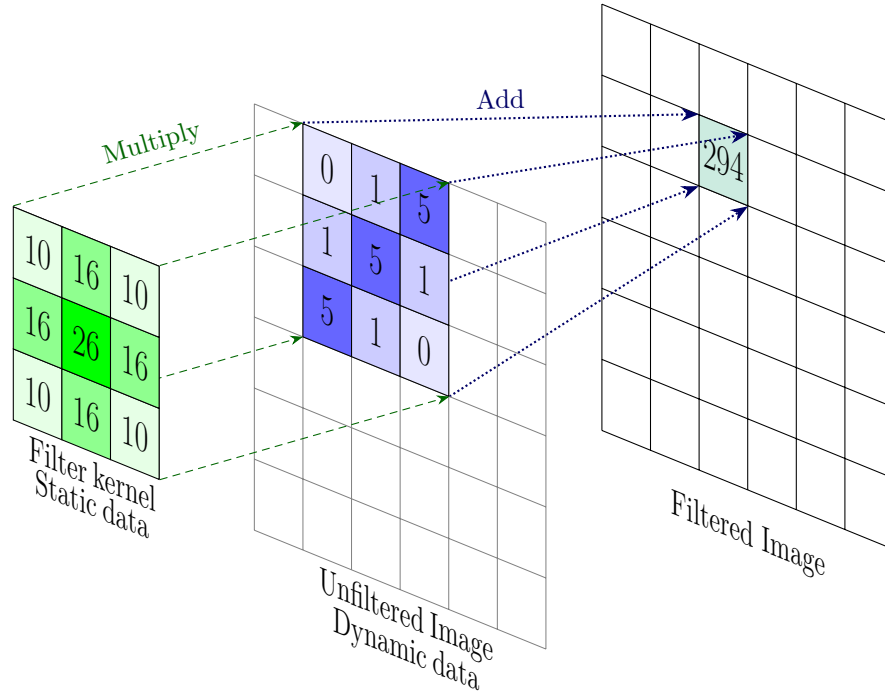


FIGURE 4.4: Schematic representation of the **FeFET**-based filter application. The filter **kernel** is stored in a non-volatile way in the FeFET. An external image is input to the circuit, resulting in a filtered image. In the case shown, the filter **kernel** values correspond to a Gaussian blur filter, and color saturation depends on pixel intensity.

Required post-processing

Another difference between 2D image convolution and 1D signal processing is the existence of possible color channels: red (R), green (G) and blue (B) channels of classical RGB images

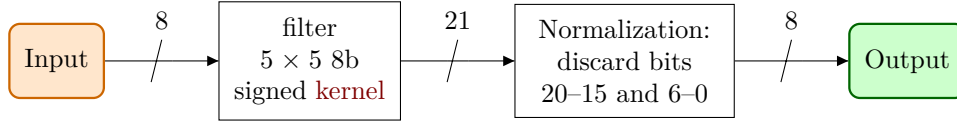


FIGURE 4.5: High-level diagram of the proposed image filter.

are usually computed individually as three distinct images with either the same, or different **kernels**. More complex color mixing can be achieved with a convolutional image filter by mixing the color components.

For a 1D convolution of M samples by a second vector of size N , the resulting vector is of size $M + N - 1$, though the result is often considered uninteresting when the **kernel** only partially covers the input signal. Discarding these values gives an output size of $M - N + 1$. This is also the case for 2D filtering, along the two dimensions of the image; though input data may require padding samples, or re-ordering if the hardware filter has no knowledge of image sizes, and/or insufficient storage capabilities.

Lastly, the convolution output must be normalized again in order for it to be in the same range as the input image: each output pixel contains the contributions of $K_w \cdot K_h$ pixels, with K_w and K_h representing respectively the width and height of the (usually square) filter **kernel**. For an image whose pixel values are in the $[0; I]$ range with a **kernel** in the $[0; K_r]$ range, the output image range is $[0; K_w \cdot K_h \cdot K_r \cdot I]$ range. For instance, for an image and a 5×5 **kernel** in the usual 8-bit unsigned ² $[0; 255]$ range, the output range is $[0; 1625625]$, which requires 21 unsigned bits to be accurately represented.

As a result, renormalization is often built inside the **kernel** coefficients for floating-point computation, or performed before storing and transmitting the image, sometimes after applying further processing steps such as gamma-transforms.

4.6.2 Filter architecture

The objective of the demonstration filter is to apply a convolution filter **kernel** to every image of an input stream, in real-time. This constrains the filter architecture, in terms of both data path (as the input data should be streamed) and timing constraints (as the operation time will directly affect frame rate).

Figure 4.5 provides a high-level view of the filter architecture.

Intermediate samples

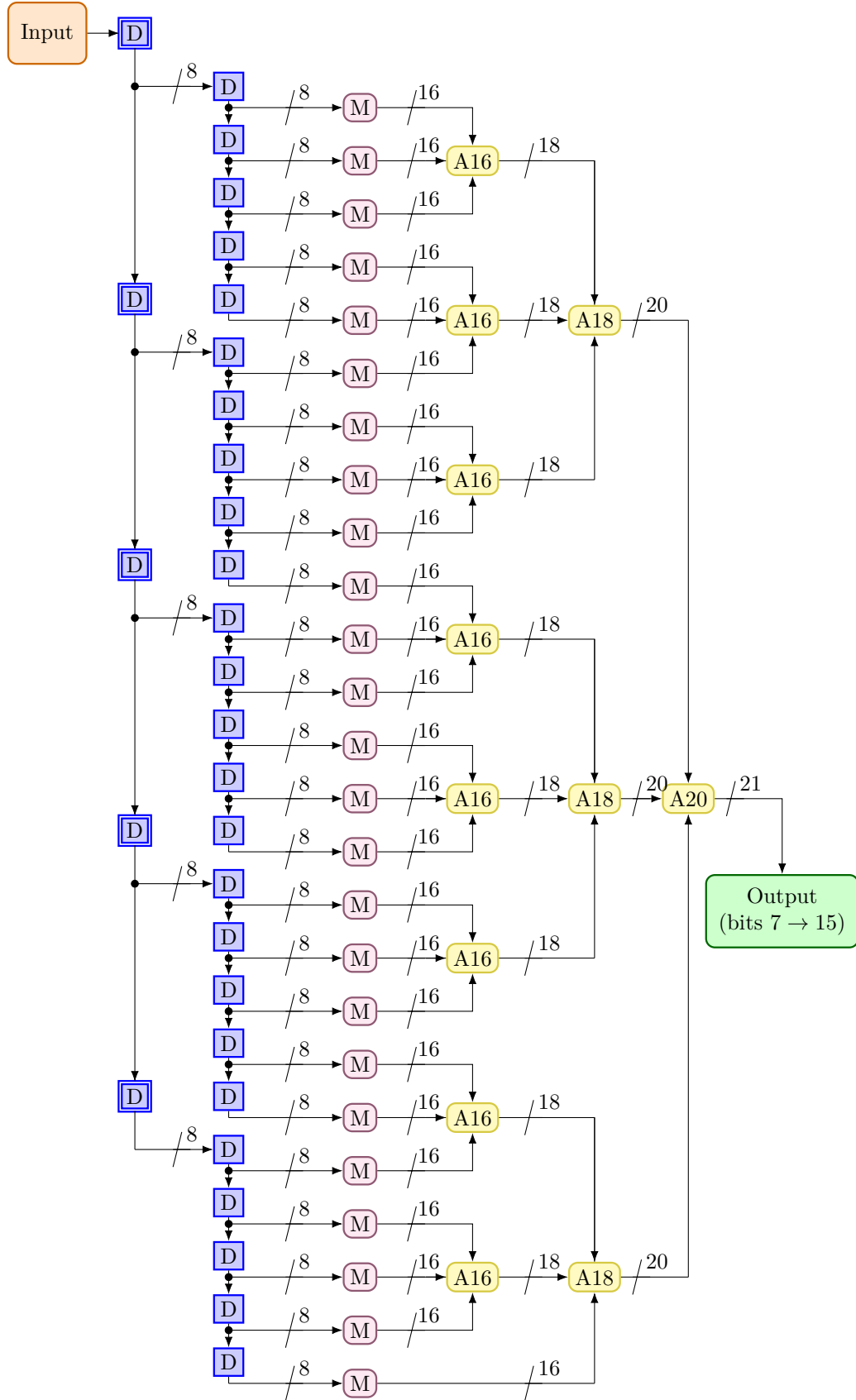
With a filtered image of width W and a square **kernel** of size K , the filtering operation processes K lines at once, with $(W - K) \cdot (K - 1)$ intermediate pixels that have to either be stored temporarily, or retransmitted later. While storing the intermediate samples was initially considered, the choice was made to simplify the design by eschewing the storage of these $(W - K) \cdot (K - 1) \approx W \cdot K$ intermediate pixel values³. This choice trades lower design complexity and area for a high bandwidth overhead, as each input pixel needs to be retransmitted K times for filtering an image.

Circuit 4.13 shows the full data path for filtered pixels. One pixel from each line of the input image is provided sequentially and stored in the first **D flip-flop** **[D]** stage. Then, the second column of **D flip-flop** **[D]** ($K_w = 5$ per image line) is activated, making the **kernel** advance laterally over the image. The multipliers **[M]** can be activated simultaneously, as they now receive the updated input values.

The input data clock therefore operates at five times the multiplier clock rate, confirming the overhead factor presented above: each individual pixel is fed five times as the **kernel** advances, the circuit having no memory of the previously seen line. This is acceptable for a demonstrator, but it can also be a viable efficiency trade-off, depending on the energy consumption of the memory. Regarding processing speed, the multiplier speed is the limiting factor in this case, as will be detailed later in section 4.6.3.

²Some processing **kernels** require negative coefficients, such as the Sobel operator. The output range becomes $[-816000; 809625]$, also requiring 21 bits; only the magnitude is often preserved

³For a full high definition image, this is $\approx 1920 \times 5 = 9.6$ kB of data, a non-negligible amount.

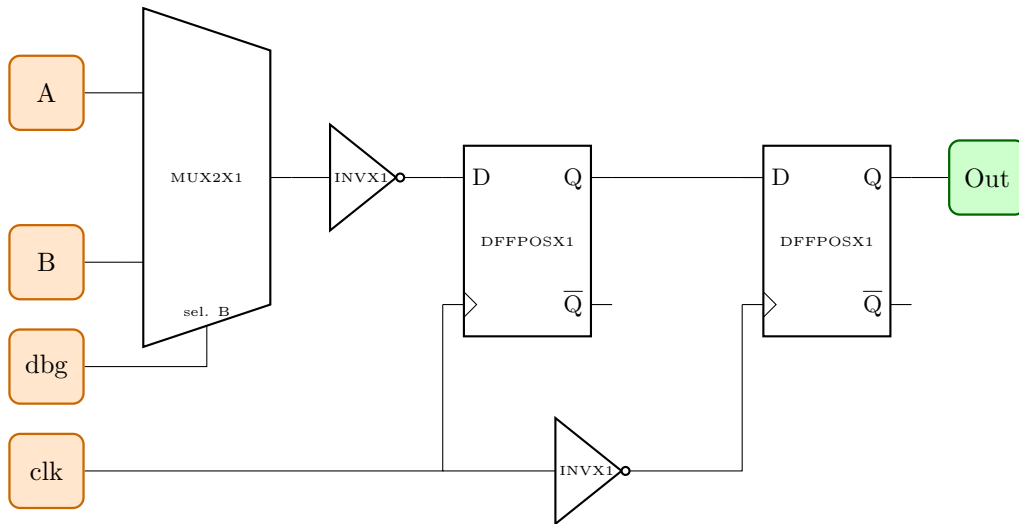


CIRCUIT 4.13: Filter architecture diagram showing the input data path without alternate scan chain. **D** flip-flops are marked **D**, multipliers **M**, and adders **Axx**, where “xx” is the input word size. **D** flip-flops **D** from the leftmost column operate five times faster than the rest of the circuit.

Alternate scan-chain

While shift registers were designed to feed multipliers with the input image data, they were modified to allow a secondary “debug” data path, where every shift register was connected serially. Moreover, additional shift registers were placed throughout the circuit, to allow sampling most signals for debugging purposes. When the debug signal is enabled, shift registers read input data from their secondary “debug” port, as visible in [Circuit 4.14](#). Initially envisioned as triggered with a falling edge, as described in [Listing A.11](#), this was shown in simulation to cause issues with the manual clock distribution, and the timing-dependent multiplier circuit. Rather than sample their input signal on a falling edge and immediately start to reflect this with their output, the circuit was altered to wait for a rising edge to change the output, as described in [Listing A.12](#).

This circuit was subsequently synthesized as a single-bit variant into [Listing A.13](#), as the layout had to be performed manually due to unavailable standard cells libraries. The single-bit variant allowed to perform the layout a single time, and use eight structures in parallel as the 8 bit shift register. An additional optimization was the reuse of a falling-edge flip-flop with an inverted clock signal, as clock propagation delays were made non-critical, and this saved re-performing the layout of the most complex part of the cell, which was the flip-flop. The corresponding design is shown in [Circuit 4.14](#).



CIRCUIT 4.14: Internal circuit diagram of the shift register. The “debug enable” **dbg** signal selects the secondary multiplexer input, which is typically connected to the previous shift register output, as visible in [Circuit 4.17](#).

Bit precision

The multiplier circuit was designed to be as generic as possible. Therefore, compatibility with common image filtering **kernels** such as the ones presented in [Figure 4.3](#) was considered during the design phase.

The requirements were the following:

- Input data: Image with 8 bit/pixel
- Output data: Image with same bit depth (8 bit/pixel)
- Filter **kernels** must accommodate:
 - Negative values (Sobel, unsharp)
 - Relatively high values, as 5×5 unsharp masks have a maximum value of 476 (9 bits)
 - A scaling factor of at least 256, to allow the use of such high coefficients

As a compromise between these requirements and design complexity with limited I/O capabilities, a 8 bit signed 5×5 **kernel** was selected.

4.6.3 FeFET-based Logic-in-Memory multiplier design

Multiplier circuit, adder circuit

Logic (N)AND gates can be implemented using a single FeFET, as seen in subsection 4.4.1. This can be directly leveraged in a multiplier circuit: as illustrated in Table 4.2, a binary multiplication computes multiple AND operations. However, the real difficulty of a multiplier circuit is the addition of carry bits.

Kernel				I_1		I_0
K_0				$K_0 \cdot I_1$		$K_0 \cdot I_0$
K_1			$K_1 \cdot I_1$	$K_1 \cdot I_0$		
K_s		$K_s \cdot I_1$	$K_s \cdot I_0$			
<hr/>						
$K_{s,ext}$	$K_s \cdot I_1$	$K_s \cdot I_0$				
	$K_s \cdot I_0$					
Out. bit	O_5	O_4	O_3	O_2	O_1	O_0
Value	$K_s \cdot I_1$ $+K_s \cdot I_0$	$K_s \cdot I_1$ $+K_s \cdot I_0$	$K_s \cdot I_1$ $+K_s \cdot I_0$	$K_1 \cdot I_1$ $+K_s \cdot I_0$	$K_1 \cdot I_1$ $+K_1 \cdot I_0$	$K_0 \cdot I_0$

TABLE 4.2: Concise example of the multiplication of a 2-bit unsigned integer (I , horizontal) with a 3-bit signed integer (K , vertical). Note the sign extension K_s : the sign needs to be extended from the last kernel bit (3rd bit K_3) to the maximum output length, 5 bits. The horizontal double bar delimits the multiplier part that does not require a sign extension, which was split into the first pipeline stage. Carry bits are not shown in the “value” expression.

As illustrated in Table 4.2, a multiplier can be considered as an adder with progressively shifted bitwise-multiplied numbers as the input operands. Bitwise multiplication is a logical AND, and can therefore be realized with a single FeFET between the value stored and the input value. A standard adder was therefore designed, with the AND logic operation integrated into a FeFET.

Ripple-Carry Adder

Due to its regularity, it was chosen to implement a ripple-carry adder to decrease design time and complexity. Ripple-carry array multipliers are very regular, and only require full-adders and half-adders, at the cost of generally lower performance and higher transistor count. Since a half-adder is also a full-adder with the carry input set to 0, it is only necessary to design a full-adder whose outputs are given by Equation 4.3 and Equation 4.4.

$$\begin{cases} S_{out} = C_{in} \oplus A \oplus B \\ C_{out} = (A \cdot B) + (C_{in} \cdot (A \oplus B)) \end{cases} \quad (4.3)$$

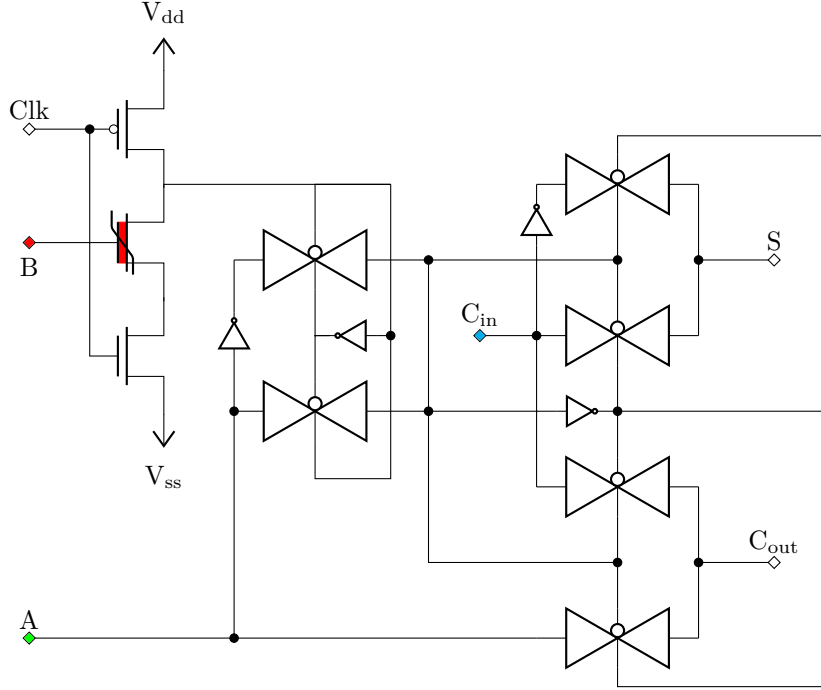
$$(4.4)$$

Circuit B.1, Circuit B.2 and Circuit 4.15 show the evolution of the full adder design across multiple successive iterations.

Pipelined architecture

As the worst-case delay of the multiplier as simulated did not allow a Full-HD 25 Frames Per Second (FPS) grayscale video feed ($clk = 1920 \times 1080 \times 25 \approx 52 \text{ MHz} \approx 49 \text{ MiBs}^{-1}$, or 19 ns/pixel) to be sustained, it was chosen to pipeline the multiplier, since the limiting factor of the first multiplier design from Circuit B.1 was the node discharge rate for the carry propagation. Pipelining introduces more design complexity, as well as a one-cycle latency penalty, but approximately doubles the throughput by allowing the node discharge to occur in parallel in both stages.

The multiplier was therefore split into two halves, as shown in Circuit 4.16 and Circuit 4.17, along the sign extension delimitation represented in Table 4.2.



CIRCUIT 4.15: Final design of a full-adder with an integrated 1-*FeFET* logic AND gate (in red), which is used in the filter application. The AND gate performs a binary multiplication of the input B with the value stored in the ferroelectric oxide before adding it with inputs A and C_{in}. C_{out} and S are the outputs of the binary full adder, corresponding to carry and sum, respectively.

The second half is slightly smaller as carry bits are not propagated further, although the implementation used twice the same circuit, while discarding extraneous C_{out} and S outputs, as shown in Circuit 4.17. Full adders are not required for the first line (and column) of the multiplier circuit, as their C_{in} and A inputs are unneeded, as visible in Circuit 4.16. While half-adders could have been designed, the same circuit was repurposed instead, with inputs set to V_{ss}, as shown in Circuit 4.17.

As a consequence of choosing to delimit the multiplier pipeline stages along the sign extension:

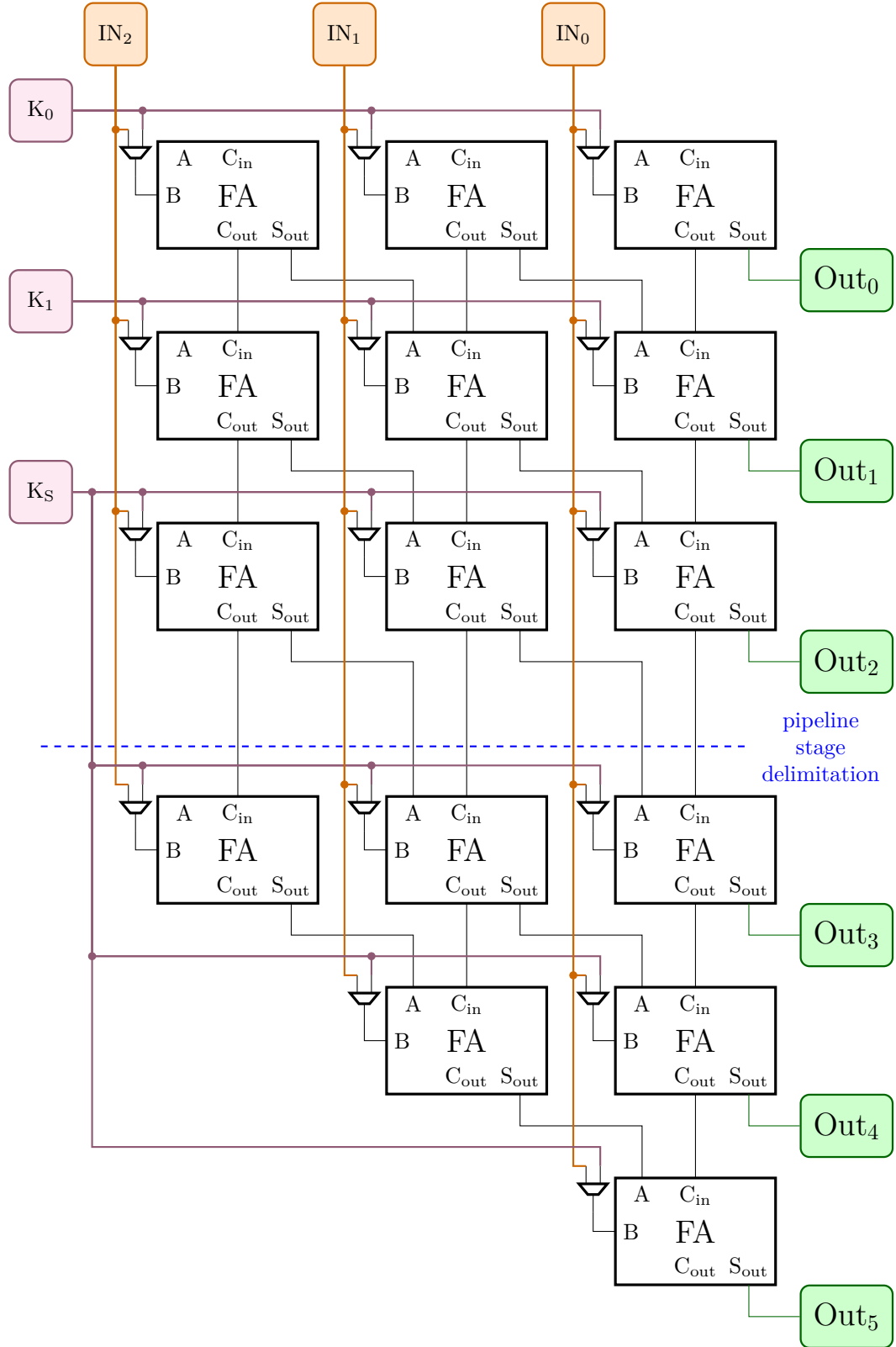
1. Each half of the multiplier outputs 8 bits (maximum magnitude output value is $-2^7 \cdot (2^8 - 1) = -32640$, which requires 16 bits)
2. Only the second half of the pipeline requires the sign extension. In fact, only the sign extension is required by multipliers of the second pipeline stage, other kernel coefficient bits are not used after the first stage.

The same 8-bit shift registers can therefore be re-purposed between the two stages.

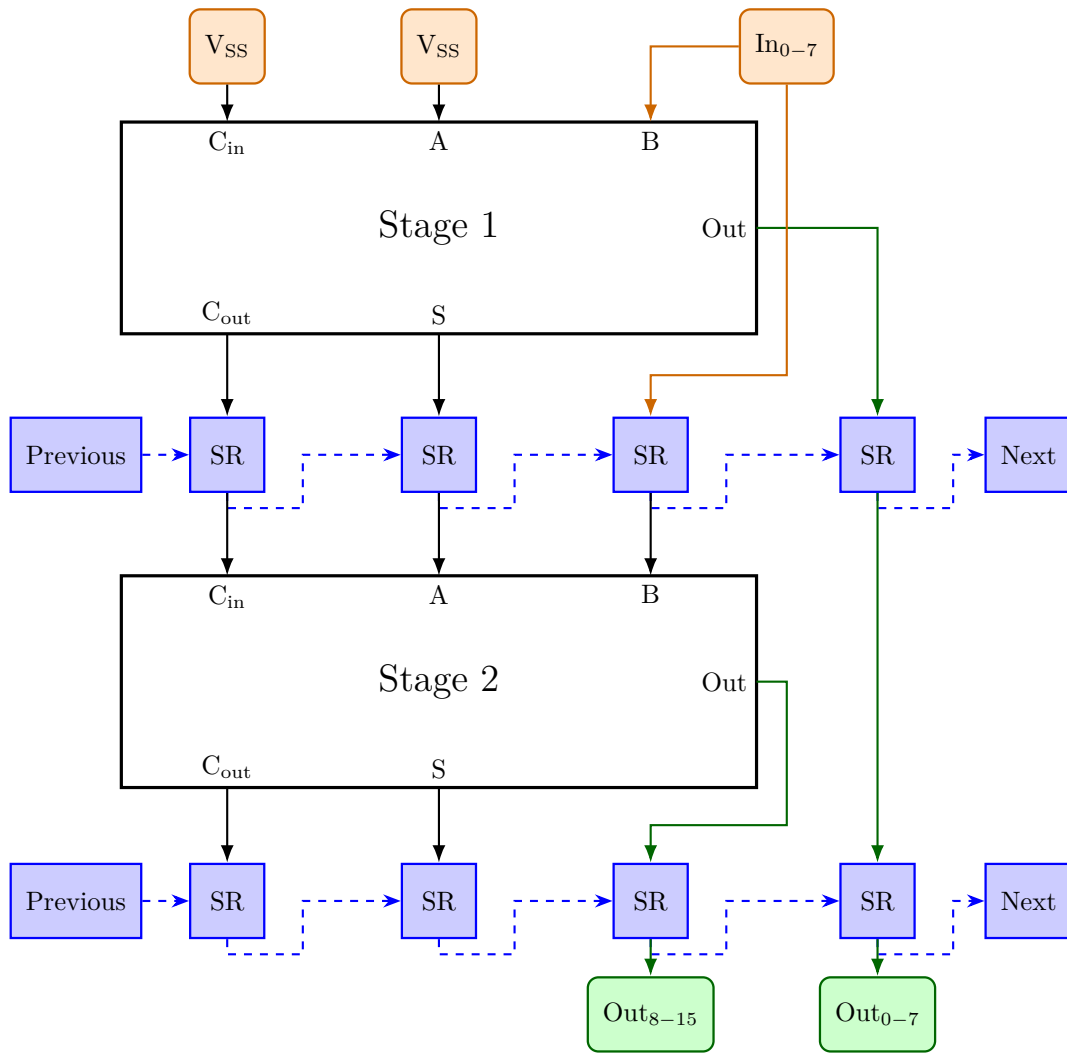
Kernel weight programming

As an added benefit of the pipelined architecture, the 8-bit filter kernel can be transmitted serially over the same 8-bit shift registers, using the alternate scan-chain: the second stage multiplier only requires 8 sign extension bits as its “filter kernel” multiplicative coefficients. The weights can therefore transit up to the exact same multiplier inputs (more precisely, the input *FeFET* in full adder circuits) that will later receive input image pixel data.

The sign extension can therefore be sent first along the alternate scan-chain, repeated in every bit of the pixel data, before sending the remaining 8 bits of the kernel data. The alternate scan-chain also connects the shift registers between multiplier stages, that are more difficult for pixel data to access during normal operation. This feeding operation for the filter data can be seen in the Verilog test-bench in Listing A.14. As the weights are stored in a



CIRCUIT 4.16: 3×3 diagram of the Ripple-Carry Adder based on full adders (fabricated version is 8×8). Indicated I/Os are image input bits IN_n , output bits Out_n , and filter kernel bits K_n . Multiplexers are controlled with the programming signal, which sets every input to the corresponding voltage-boosted kernel bits. Doing so writes kernel coefficients in every full adder. The critical chain follows the leftmost full adder of each row, unconnected inputs are pulled low.



CIRCUIT 4.17: Multiplier pipeline stages, with their **inputs** and **outputs**. This schematic shows the full pipelined multiplier architecture with the 8 input bits, and 16 output bits, as well as the **shift registers** used for pipelining, including the alternate scan-chain used for debugging and programming. The last multiplier has the post-stage 1 alternate scan chain connected to the first multiplier's post-stage 2. Each data line represents an 8 bit bus.

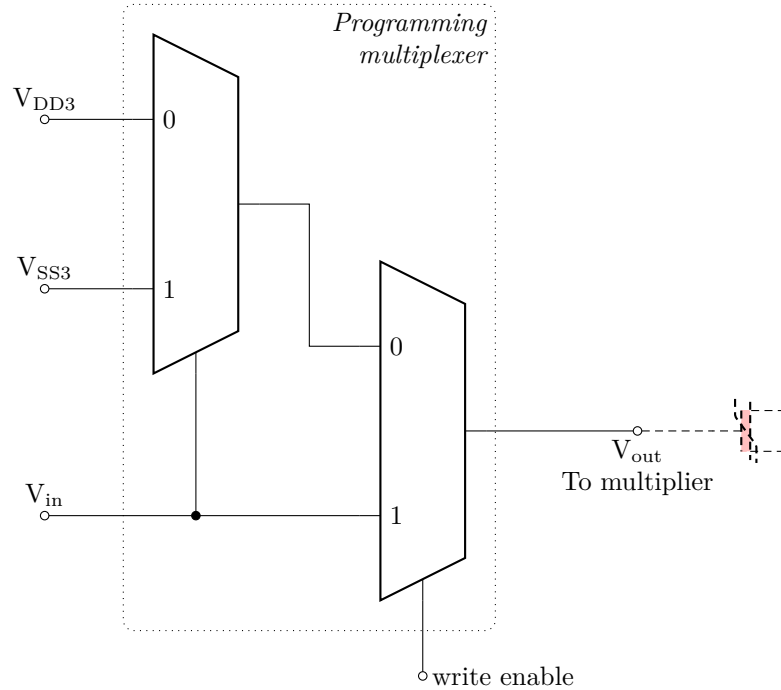
This schematic represents block M in **Circuit 4.13**.

non-volatile manner, this operation is not time-critical, and can therefore be implemented in a more space-efficient way, as with the shift-register approach outlined here.

It should be noted that this approach stores the sign extension bits separately, instead of automatically extending the sign bit of the kernel. This approach allows performing unsigned operations as well (by using zero-valued sign extensions), or to program larger kernel values for unsigned operation: despite the overflow risk, this may be leveraged for computing.

Once weights are “aligned” with the corresponding multiplier, meaning each kernel bit is positioned where image bits would be during filter operation, the bits must first be re-distributed: as shown in [Circuit 4.16](#), each image bit is distributed over a column of **full adders**, while kernel bits are distributed over lines. This is performed by activating a multiplexer in front of each **full adder**, controlled by the *write enable* signal. In order to program input **FeFETs** with the presented kernel weights, voltage levels also need to be raised above V_C in order to polarize them. To do so, a voltage level shifter is positioned after the multiplexer. As shown in [Circuit 4.18](#), this circuit is also composed of two multiplexers, to raise or lower the input value to the correct high-voltage value. When the programming signal is active, the input voltage is raised to either of the externally provided programming voltages V_{DD3} and V_{SS3} , shown as a timing diagram in [Figure 4.6](#). Note that, due to concerns detailed in [section 4.6.4](#), the *write enable* signal is effectively the same as the “debug” signal that activates the alternate scan-chain. This is not an issue during normal operation of the alternate scan-chain, as the ferroelectric oxide will not be re-polarized while the power supply voltages V_{DD3} and V_{SS3} remain under the V_C threshold.

While it should be possible for both V_{DD3} and V_{SS3} to simultaneously output their respective $\pm V_C$ values, this would create potential differences of up to 6 V in this case, which could damage even the input/output cells. Instead, as the programming operation timings are not critical, the voltages are staggered, as shown in [Figure 4.6](#): **FeFETs** that need to be programmed with a “logic high” state are programmed first, and “logic low” **FeFET** are programmed during the second pulse. This scheme could also possibly lower peak power consumption during writing. The timing of the actual programming pulses is discussed in [section 4.6.5](#).



CIRCUIT 4.18: Voltage multiplexer used for programming the multiplier **FeFET**. Depending on the V_{in} and *write enable* inputs, the multiplier will either receive the V_{in} signal (during normal operation), or one of the V_{DD3} and V_{SS3} voltage levels during programming. This circuit was implemented by **NaMLab**, the second multiplexer with two transmission gates.

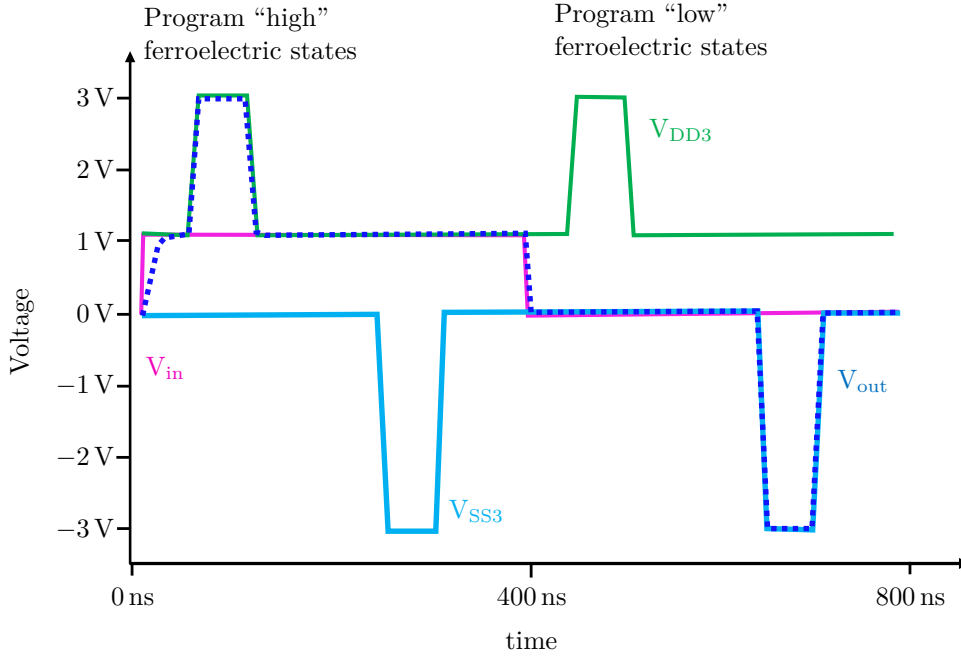


FIGURE 4.6: Input and output of the multiplexer described in [Circuit 4.18](#) and connected to the input “B” FeFET of the full adder pictured in [Circuit 4.15](#). When programming is enabled, the multiplexer will boost the V_{in} signal to V_{DD3} or V_{SS3} , as visible on its output V_{out} .

4.6.4 Validation in simulation and identified issues

The main contribution to the design made by NaMLab was to check it in simulation: model instability, as well as the size and complexity prevented running the complete circuit in simulation.

Simplifying the ferroelectric model, creating a Register Transfer Level (RTL) representation of the circuit as well as a software implementation, and comparing results at each step contributed to the identification of two major flaws in the initial design, that would have prevented the circuit from working.

Simplification of the circuit-level simulation

The proposed filter contains 25 multipliers in order to operate with 5×5 kernels, each storing one full pixel of the filter kernel in binary representation. Each multiplier has two pipeline stages, each with an 8×8 matrix of full adders containing 1 FeFET each. The total count is therefore $8 \cdot 8 \cdot 2 \cdot 25 = 1600$ FeFETs, and a few orders of magnitude more transistors.

Simulating such a circuit is feasible, however:

- memory consumption of the simulation is relatively high, between 10 GB to 20 GB;
- simulation speed is an issue, taking a few days per kilopixel on an 8-core machine;
- model instabilities make convergence difficult.

The first two items can be improved on by using more powerful hardware for simulation. However, the experimental ferroelectric model used to model FeFET exhibited rare convergence issues, which were exacerbated by the number of devices present in the circuit, preventing successful simulation of even a few clock cycles. Since the model was also relatively slow and memory-intensive due to keeping track of the ferroelectric history and the turning points as described in [subsection 2.2.2](#), replacing it with the simplified model from [Listing 2.1](#) described in [subsection 2.2.3](#) improved on every point, allowing successful simulation of the image filter processing a full $30 \text{ pixel} \times 30 \text{ pixel}$ image in a few hours.

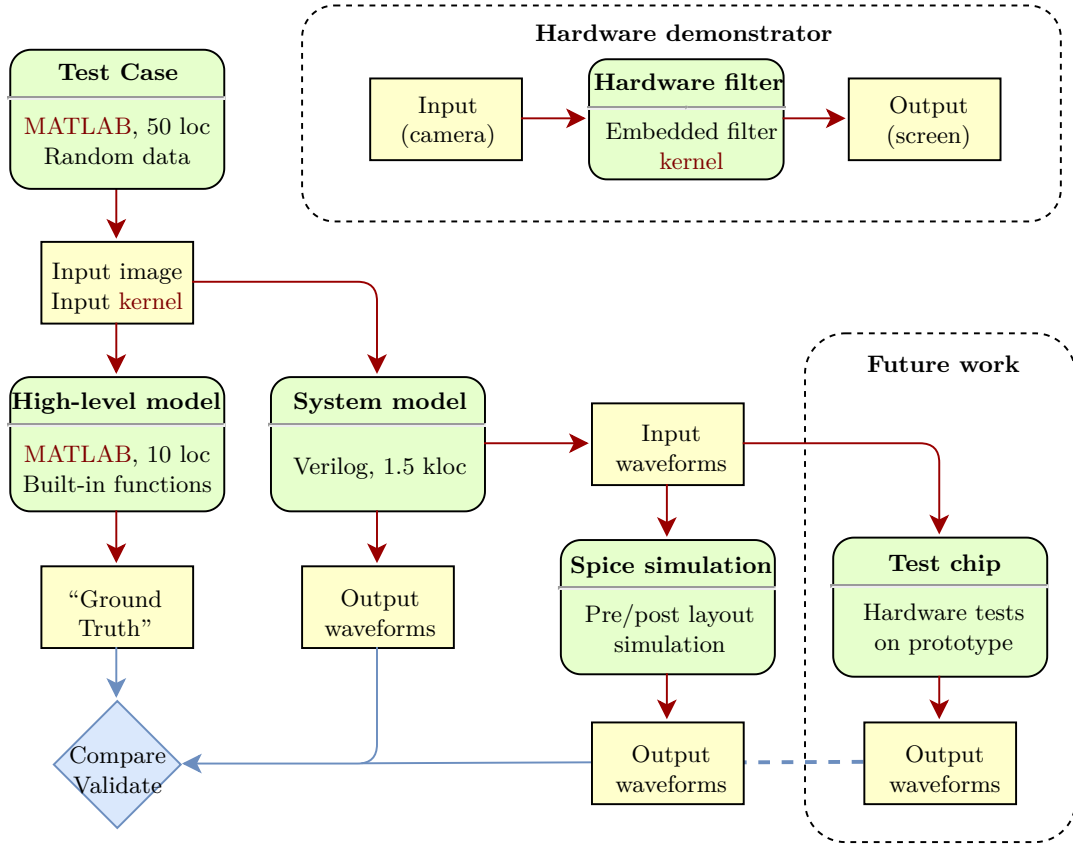


FIGURE 4.7: Verification flow employed to validate the image filter design

Generation of reference input and output signals

To make verification of the output possible, it is necessary to construct a known-good reference. A top-down approach was used for generating output data corresponding to arbitrary (pseudo-random) input data. The objectives were to:

1. validate that the logic operation performed by the circuit was exactly the same as that defined in image processing libraries
2. aid the creation of complex test patterns for programming the filter **kernel**, then feed the input data, while respecting relative clock timings, signal dependencies and relative alignment of the signals
3. obtain reference signal outputs to be compared with circuit simulation outputs.

To do so, a multi-stage testing approach was devised, as represented in **Figure 4.7**.

Firstly, a very high-level model of the filter was created using the built-in **conv2** 2D convolution function in **GNU Octave** (MATLAB-compatible), as well as random input test data. The output is considered the “Ground Truth”, and other models have to match this output to be considered functional.

Secondly, a high-level **Verilog** model was created, with the objective of creating a cycle-accurate representation of the input and output of the image filter. This model was developed while undergoing continuous functional testing to ensure it remained compatible with the theoretical output. However, as progress was being made in parallel both on the circuit schematic and the **Verilog** model, it became evident that a black-box model of the filter was neither useful for debugging, nor easy to tune to match the schematic: as differences started to appear in the output, modeling of subcomponents became a necessity to isolate the faults. The final **Verilog** model hierarchy thus closely resembles that of the circuit design.

Thanks to the availability of a detailed **Verilog** model, it is possible to compare internal signals at various points. Moreover, the availability of an alternate scan-chain enables an

automatic test pattern generation-like approach, by generating pseudo-random input, simulating it in the various models, and comparing the outputs. “vcd” waveforms generated from the Verilog testbench simulated under Icarus Verilog[Wil] were directly imported inside the Cadence Spectre simulator⁴. Verilog-A blocks such as those listed in Listing A.5 and Listing A.6 were used to serialize data transiting on internal buses to binary files that could directly be compared with the ones emitted by their Verilog counterparts. Automating the above comparison proved very valuable in diagnosing two problems detailed in the following sections.

Interestingly, all the uncovered issues were integration issues that could not have been discovered by running isolated simulations. These findings illustrate the use-case for simplified models in large-scale simulations.

Wrong clock trigger for multiplier

This first issue is relatively simple, but only a simulation encompassing both a multiplier and its input shift registers can highlight it. In the first designed version of the filter, multipliers shared the same clock as the shift registers latching their output. That caused a timing issue where the shift registers received a rising edge to store the multiplier’s output before they finished computing. This only occurred for the slowest multipliers (that is, in cases where the carry bit needs to be propagated across multiple full adders).

The obvious fix would have been to adjust the relative delay between these two clocks, but getting timing wrong would also mean a non-functioning circuit, so using a dedicated clock signal for the multipliers was preferable. However, the pad-limited design made adding an extra signal difficult, the 25 pad budget being fully utilized.

More interesting is the alternate writing scheme used to free up the `write_enable` pad for an extra clock signal. Pad assignment was previously, besides the 16 pads taken by 8 bit parallel input and output signals:

- regular power supplies: $V_{DD} = 1.0\text{ V}$, $V_{SS} = 0.0\text{ V}$;
- ferroelectric programming supplies: $V_{DD3} = V_P$, $V_{SS3} = -V_P$, with V_P the chosen programming voltage, greater than V_C ;
- an additional voltage V_{shift} for driving the transistors in the level shifter for the programming circuit shown as the left multiplexer in Circuit 4.18;
- Clk_1 as the clock used for the first five shift registers in the input data path shown in Circuit 4.13, nominally five times the rate of Clk_2 ;
- Clk_2 used for multipliers and shift registers;
- $\text{Write}_{\text{Enable}}$ as a control signal for boosting voltage levels of ferroelectric transistor inputs from V_{DD} and V_{SS} to V_{DD3} and V_{SS3} levels respectively, to allow repolarization of the ferroelectric layer;
- $\text{Debug}_{\text{Enable}}$ as a control signal for enabling the alternate “debug” scan chain, where shift registers are serially connected.

The $\text{Write}_{\text{Enable}}$ signal was merged with the $\text{Debug}_{\text{Enable}}$ signal, and externally supplied programming voltages were made dynamic, as these need to be high enough to make writing possible:

- Unchanged: V_{DD} , V_{SS} , V_{shift} , Clk_1 , Clk_2 ;
- V_{DD3} and V_{SS3} are now externally controlled to either regular V_{DD} and V_{SS} voltages, or $\pm V_P$ during programming.
- Added Clk_3 a new clock signal used for multipliers, using the pad previously used for $\text{Write}_{\text{Enable}}$;
- $\text{Debug}_{\text{Enable}}$ is now used both to trigger serial debugging mode, and to boost both V_{DD} and V_{SS} to levels provided by V_{DD3} and V_{SS3} , respectively.

⁴In ADE L, Setup -> Simulation files

Since the programming voltages are supplied externally, they can be lowered to regular logic voltages if programming the ferroelectric oxide is undesired. As such, $\text{Debug}_{\text{Enable}}$ now activates the same level shifters as $\text{Write}_{\text{Enable}}$ previously did, but the side effect of repolarizing the ferroelectric oxide will can be avoided by lowering V_{DD3} and V_{SS3} below V_C .

Node discharge dependency

The other critical issue uncovered led to a complete re-design of the full-adder and fused **FeFET**-based multiplier. This is an issue that could only be uncovered in a post-integration simulation.

The core issue stems from the use of dynamic logic due to p-**FeFET** unavailability: a capacitive line is pre-charged, then the expression is evaluated, and the line is discharged if needed. However, that line cannot be brought up again before the next clock cycle, even if the inputs change. Such a dependency issue appeared in simulation, and led to the complete redesign of the then fully-dynamic logic-based multiplier into a mixed dynamic and static logic design.

Circuit 4.16 illustrates the issue: the carry bit ripples through the adders, causing large worst-case latency. When the latency is higher than the time left for evaluation, the carry bit is not considered when computing the output value, causing errors in the most significant bits.

The original circuit pictured in **Circuit B.2** is composed of two dynamic halves: the first, on the left, computes the carry output C_{out} from three inputs, among which the carry C_{in} from the previous digit. The resulting carry feeds into the next dynamic part by way of the $\text{NOT}(C_{\text{out}})$ floating node. In case that node begins discharging prematurely due to other circuits not having yet computed their output carry, it cannot be brought back up, and both the carry and bit output of the stage will be wrong.

This is fixable by carefully delaying the clock signal to start evaluating the most significant bits only after waiting for the worst-case carry ripple latency, however that also means intentionally limiting the circuit speed by setting a minimal latency. Moreover, precise circuit latency was not known for this technology pathfinder, and introducing a minimal latency would have prevented accurate latency measurements.

Instead, the circuit was re-designed using a mixed **CMOS**-and-dynamic logic approach, as described in **section 4.3.3**. The final design presented on **Circuit 4.15** makes use of a single clock-gated ferroelectric transistor, whose output controls multiple transmission gates. These transmission gate control signal are the only floating nodes, and their state does not depend on the input carry, as that part of the computation is controlled using conventional logic levels. That eliminates a dependency on the previous multiplier stage, which in turn relaxes timing constraints, allowing for lower latency computations. The total number of transistors is increased, but the number of **FeFETs** is reduced. **FeFET** being larger ($500\text{ nm} \times 500\text{ nm}$ in this case) than regular transistors, this leads to comparable circuit geometries. Constraints on line capacitance for the precharged floating node can be relaxed too, permitting the use of smaller transistor gates connected to the floating node, which also accelerates computation speed.

As a result, the design was improved by reducing the number of **FeFETs** as well as using as little dynamic logic as the circuit allowed. This also improved the maximum propagation delay in the multiplier, reducing it to 1.52 ns in simulation, making the pipelined architecture unnecessary. The subsequent adders only necessitate 535 ps , 595 ps and 661 ps each, for the 16 bit, 18 bit and 20 bit version, respectively.

4.6.5 Results

Weight programming

When programming weights according to the scheme described in **section 4.6.3**, two values can be tuned, which will influence the threshold voltage: programming pulse height (voltage), and programming pulse width (timing). Both influence the number of domains that will undergo repolarization: the higher these values, the larger the V_{th} shift will be.

Experimental measurements are displayed in **Figure 4.8**. In light of these measurements, a programming voltage of $\pm 3\text{ V}$ was chosen, in order to preserve the hardware, with a pulse

time of 10 μs , as programming is not a time-critical operation, and is only performed relatively infrequently, when kernel coefficients are updated. Moreover, positive and negative voltages are not supplied simultaneously to avoid 6 V potential differences, which would be hazardous to transistors, including the reinforced Input/Output ones.

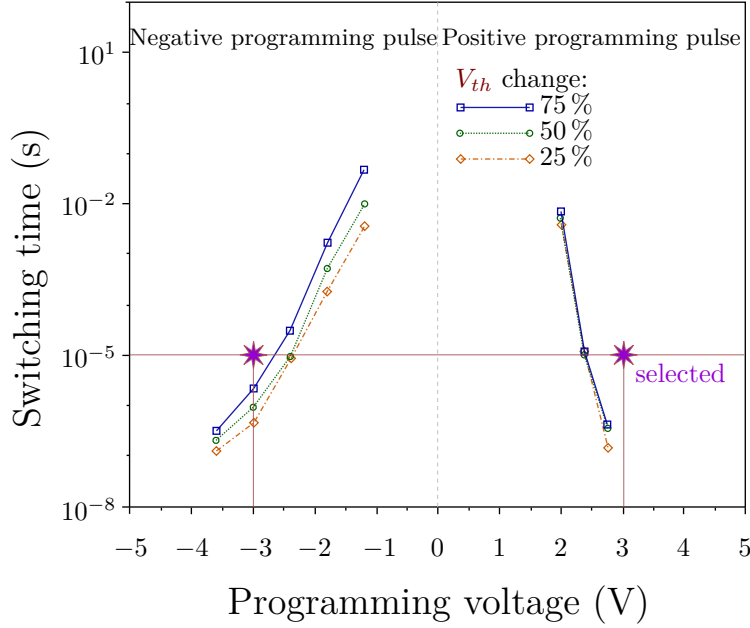


FIGURE 4.8: Measured pulse time required to achieve a V_{th} change of the FeFET by 25 %, 50 % and 75 % between low- V_{th} (V_{th}^{FL}) and high- V_{th} (V_{th}^{FH}) as a function of pulse voltage. Stars indicate the position of chosen program and erase timings (10 μs) for 3 V operation. These operating points were chosen to obtain the largest possible V_{th} change while preserving the circuit from high voltages.

Operating point

As visible on [Circuit 4.15](#), the ferroelectric transistor's drain-source resistance controls the discharge rate of the floating node. That resistance therefore requires careful tuning, as the floating node must not discharge before the carry has been propagated to every **full adder**: $R_{DS}(\text{off})$ must high enough. The same node must also discharge relatively quickly in the opposite case, as this input is necessary to compute the carry that propagates. We therefore have four cases:

A (V_{in})	Ferro state	R_{DS}	Shifted V_{th} expression
High	High	Low	$V_{in}^H + \Delta V^+ > V_{th}$
Low	High	High	$V_{in}^L + \Delta V^+ < V_{th}$
High	Low	High	$V_{in}^H - \Delta V^- < V_{th}$
Low	Low	High	$V_{in}^L - \Delta V^- < V_{th}$

According to the above table, five parameters can be adjusted to respect the conditions:

- V_{th} : Depends on the fabrication process, intrinsic to the **FeFET**
- V^+ and V^- : V_{th} shift in both directions, depending on weight programming procedure
- V_{in}^H and V_{in}^L : logic voltages (V_{DD} and V_{SS}) for circuit **I/O**

The first point is a process optimization step, more than a design tunable parameter. It can be further adjusted with doping or gate work-function engineering[Rei+19]. The second point was chosen to maximize the V_{th} shift ΔV , as detailed in the previous part. Finally, the third point can be relatively freely adjusted, with the caveat that this will affect **CMOS** performance as well.

Figure 4.9 shows the data gathered in order to choose the operating point. With known filter inputs and **kernel** weights (both either all zeroes or ones), the output is observed, and compared to the expected one.

An early discharge, more likely to occur when the input is close to the programmed **FeFET** threshold voltage $V_{th} + \Delta V_{th}^x$, occurs when the output unexpectedly becomes “1”. This is shown in light blue in Figure 4.9. As expected, such an event is more likely when the ferroelectric oxide is in the “high” (low- V_{th}) state, or with input high. On the contrary, dark blue areas show where that discharge is expected. Light red areas show unexpected zeroes in the output, and dark red expected zeroes. In between are areas marked grey where only some output bits changed to zero or one. These areas always represent unwanted outputs.

Successful operation of the image filter must therefore occur in a region where only expected outputs are present. This corresponds to the union of darker shades, and is represented in green on the right of the diagram. An operation point at 0.6 V and around 22 ± 0.5 ns was identified, just short of the 19.2 ns required for Full-HD processing at 25 **FPS**, which highlights how difficult the tuning of such circuits is. The circuit is still usable at a lower frame-rate of 23.5 **FPS** to 24.7 **FPS**.

As visible on the graph, another issue is that the “invalid” gray area is generally much wider for the **Most-Significant Bit (MSB)**(digit 0) than for the **Least-Significant Bit (LSB)** (digit 7). This is due to the fact that multiple **full adders** contribute to the **MSB** (longer critical path for the carry) while a single one contributes to the **LSB**, as pictured in **Circuit 4.16**. Device-to-device variability implies that some multipliers have floating nodes that discharge faster than others, so combining more multipliers will therefore result in a larger range of timings producing invalid outputs.

Also visible on this graph is how lower operating voltages result in slower device operation, due to both lower **FeFET** and evaluation **n-MOS** gate overdrives. This will also slow down **CMOS** operation speed due to higher **n-MOS** $R_{DS}(ON)$.

Complete interactive demonstrator

The image filter was also demonstrated with an hybrid platform combining the filter **ASIC** with an **FPGA** implementing additional logic. The **FPGA** decodes frames from an **HDMI** input, feeds them to the image filter through its **GPIO**, and retrieves the filtered data through the same **GPIO** interface. It therefore handles the necessary buffering that was eschewed from the **ASIC** design, thereby also providing more flexibility. The **FPGA** also handles clock signal generation, and programming of the various filter **kernels**.

4.7 Conclusion

4.7.1 **FeFET**-based logic

FeFET-based logic has the potential to accelerate logic operations by storing one of the operands in-situ: inside the logic circuitry itself, in **Non-Volatile Memory**. This is likely to be useful in specific cases, and comes with a set of constraints, mostly due to the extra logic necessary to reprogram transistors.

Indeed, ferroelectric transistors effectively need a separate addressing circuit to provide them with the values to be stored. Programming of ferroelectric oxides is voltage-controlled, which requires a level shifter. Progress has been made towards reducing the required voltages, though higher values remain intrinsically required to differentiate programming signals

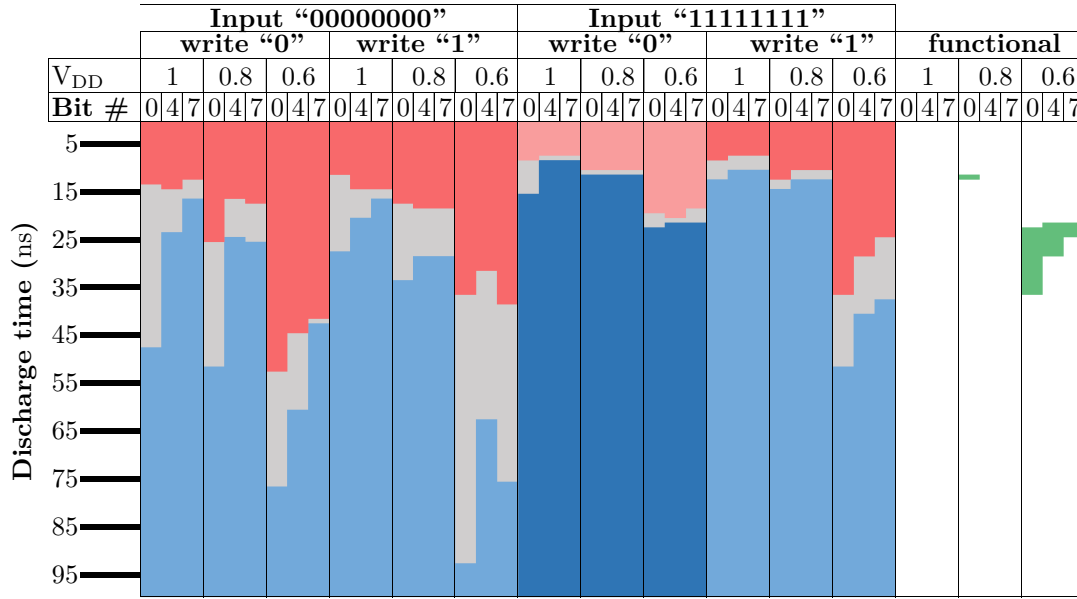


FIGURE 4.9: Dynamic characterization results of the image filter design. Two input patterns of representing either a fully black "00000000" or white "11111111" pixel are fed to the multiplier input. For each of these two cases, the case where the ferroelectric oxide was programmed with a positive or negative pulse is examined, and the output value of each bit is plotted. Changes in the multiplier output allow measurement of the floating node discharge time. The output is observed: all-one outputs are plotted in blue, while all-zeroes are visible above, in red. Mixed outputs are in the middle, in gray. Dark regions highlight the expected output, the operating point therefore has to be chosen at the intersection for proper filter operation. The corresponding voltage and timing values are displayed in green on the right. Each data point represents the statistical result of 100 multiplication operations.

from logic values. In turn, the rest of the circuit needs to be compatible with these higher programming voltages.

The need for higher voltages could be lessened with an additional access transistors for programming the ferroelectric layer, as detailed in [section 3.5](#) with the 2T1C circuit. Moreover, the use of 2T1C or related **PsFeFET** structures allows the ferroelectric oxide to be shared across multiple transistors, which is particularly interesting when considering **CMOS** circuits, as p- and n-**FeFET** receive the same signals.

Unfortunately, separating the ferroelectric layer from the transistor gate increases leakage currents from the floating node, thereby lowering the retention capabilities of the device. If longer retention periods are unneeded, most circuits described in this chapter can also be achieved with a regular paraelectric capacitor in series with the transistor gate, using a **FGMOS**-like structure, since the ferroelectric polarization reversal is not leveraged during normal use. The floating node still has to be charged, which could be done via a second transistor as with the 2T1C structure, or through other means (**flash** memory uses hot carriers, which requires impractically high voltages).

For now, it remains to be seen on a case-by-case basis whether the added complexity of including **FeFETs** in a computing circuit is worth the efficiency trade-off.

4.7.2 Image filter

As a more specific case study, a promising application was investigated, whose design capitalizes on the **FeFET** strengths (non-destructive read, non-volatility), while avoiding their weaknesses (more complex writes).

By focusing on a circuit that manipulates a sequential input stream such as video, audio, or other sensory data and processes by performing direct **LiM** operations with **FeFET**-stored data, their long data retention capability is leveraged, while avoiding frequent writes with their associated parasitic, and possibly damaging charge trapping effects. As the programming phase is not performance-critical, this use-case avoids the read-after-write issue, while still making use of the reconfigurability properties. This also simplifies the programming logic, though it does not showcase the high speeds and low-energy writes **FeFET** are capable of.

This image filter implementation demonstrates the feasibility of using **FeFET** in realistic compute workloads, leveraging their performance and energy efficiency.

The demonstrator also highlights multiple pain points, including the narrow operating margin of the device, relatively high **FeFET**-to-**FeFET** variability, and the hopefully temporary design limitations due to the unavailability of p-**FeFET**. Nevertheless, it provides an interesting benchmark of current **FeFET** technology, and an optimization target with which to measure progress in the technology process.

A regular filter design would likely store weights in volatile **SRAM**-like registers or latches, and these would need to be set to the wanted coefficients after every power loss. In contrast, the non-volatility of the presented design has potential to reduce delay and energy costs associated with start-up, as well as static power consumption. Assuming the filter coefficients rarely change, constraints on the writing mechanism can be lowered (slower data path and longer programming times are tolerable), possibly reducing the impact on the rest of the circuit.

4.7.3 **FeFET**-based memories

The same considerations apply for memories, where retention characteristics of ferroelectric oxides make them appealing, with projected retention periods of over 10 years, as well as fast, low-energy, moderate-voltage read and write operations. The baseline comparison in this case is with **flash** memories. With better retention, possibly greater endurance and faster operations, lower programming voltages and energy consumption, as well as **CMOS** compatibility, **FeFET**-based memories are well positioned against **flash**-based memories.

There may however be performance gains to be achieved using hybrid architectures, as **FeFET** devices combine two retention mechanisms: ferroelectric (exploited in 1T1C) and capacitive (as used by **flash** and **DRAM**). Adding an access transistor to the floating gate could allow the use of both mechanisms semi-independently, possibly increasing storage density while increasing performance metrics by leveraging the strong points of each. As discussed in

section 3.5, this would also improve their programming characteristics, by lowering necessary voltage levels, and improving their write endurance.

Further density gains may also be made possible with denser, NAND flash-like structures.

Chapter 5

Design space exploration and optimization

Contents

5.1 Introduction to design space exploration	114
5.1.1 Parameter space and performance space, Pareto optimal	114
5.1.2 Tool-assisted exploration	115
5.1.3 System-level benchmarking	117
5.2 Design space exploration tools	117
5.2.1 LIFT optimizer	117
5.2.2 Cadence IPC	118
5.3 Design space exploration results	119
5.3.1 Sampling of 1T1C bitcell design space	119
5.3.2 Non-volatile FeFET-based NAND gate (NV-NAND2)	125
5.4 System-level benchmarking Platform	129
5.4.1 Introduction	129
5.4.2 Scope of the benchmarking platform	129
5.4.3 Implementation	130
5.4.4 Operation modules and model cards	133
5.4.5 Example case: Adder	134
5.5 System-level exploration results	135
5.5.1 Normally-off use-cases	136
5.5.2 Interpolator simulations	137
5.5.3 Matrix multiplication benchmark	138
5.6 Conclusion	139
5.6.1 Design-Space Exploration	139
5.6.2 System-level benchmarking platform	140

In order to explore the possibilities afforded by FeFET-based circuits and LiM architectures in a cost and time-effective manner, it is necessary to go beyond the current performance levels of hardware, both in terms of device characteristics and in terms of hardware complexity. Therefore, a simulation-oriented approach was undertaken, under which a scalable benchmarking platform was set up for use in both 1T-1C and FeFET-based DSE.

The goal being to target a Design-Technology Co-Optimization (DTCO) approach in order to bridge the gap between device-level performance characteristics and system-level performance metrics while considering architectural particularities.

The objectives of this exploration are to:

- understand how normally-off computing approaches can benefit from ferroelectric memory arrays with respect to other non-volatile memory technologies and to evaluate the impact of system-level parameters (duty cycles, embedded logic complexity, sensor source activity, etc) and device-level parameters (endurance, read/write power/time, etc) on overall performance.

- explore coarse-grain **LiM** applications, including the implementation of low-level logic functions (**full adder**), a library of signal processing functions optimized for speed and for energy-efficiency, and the implementation of a complete image pre-processing application.
- explore fine-grain **LiM** designs through the development of a library of non-volatile logic gates, that can be comprehensively benchmarked with quantitative performance comparison to other non-volatile device technologies.

5.1 Introduction to design space exploration

Most circuit components require elaborate models for accurate simulation, each having multiple degrees of freedom along the possible values of their parameters. This is especially true of ferroelectric models because of their hysteretic behavior. Combining more than a few devices into more complex circuits thus makes exploring the whole parameter space infeasible. Yet, their interaction is often integral to the functionality of such circuits.

Some parameters are controlled by the designer, while others are subject to variability, up to device-to-device variations. **DSE** tools aim to ease the design task by automating the selection and simulation of parameter sets, to obtain corresponding performance results. An exhaustive exploration of this parameter space and the associated results, which form the design space, is impossible due to combinatorial explosion, as illustrated in **Figure 5.1**: as the number of devices increases, the number of possible parameter combinations rises polynomially, together with the interconnection possibilities[Poi22, p. 73]. **DSE** tools must therefore use heuristics to explore a subset of the design space that is maximally useful to the designer. Having these data points at their disposal, designers can make more optimal choices, explore the trade-offs, and better predict the system's behavior.

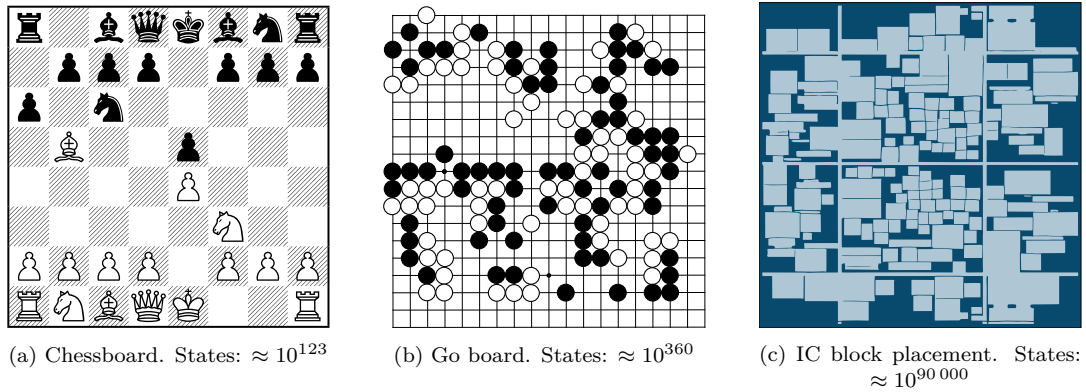


FIGURE 5.1: Illustration of complexity increase due to combinatorial explosion[Syn21].

5.1.1 Parameter space and performance space, Pareto optimal

Parameter and performance space

To characterize the behavior of a circuit, device or system, it is necessary to know its properties. Some of these properties can be controlled by the designer, such as the geometry of some transistors, or they can be an intrinsic characteristic of the material, or fabrication process. Some variability inevitably occurs during fabrication, and can also be accounted for.

The goal of a designer is to choose a set of controlled parameters that will allow the device to function despite potential external factors, while maximizing performance. Properties that are controlled by the designer form the parameter space \mathcal{X} . The range of achievable performance characteristics constitutes the performance space \mathcal{F} . This is illustrated in **Figure 5.2**.

A given set of parameters x corresponds to a set of performance characteristics f . It is important for the designer to define *performance metrics*, as numerical values used to

quantify performance characteristics. Finding parameters that maximize performance can then be expressed as an optimization problem to minimize these numerical values.

Translating parameters into metrics is carried out via the transfer function $F : \mathcal{X} \mapsto \mathcal{F}$. This function represents the behavior of the fabricated device, or its approximation using a simulation model.

Multi-objective optimization

Multi-objective optimization is generally expressed as:

$$\min f = F(x) = \begin{cases} f_1(x) \\ \vdots \\ f_n(x) \end{cases} \quad \text{constrained with} \quad \begin{cases} c_{eq}(x) = 0 \\ c_{ineq}(x) \leq 0 \end{cases} \quad (5.1)$$

where c_{eq} , c_{ineq} represents all application constraints, formulated as equalities and inequalities, x is a set of parameters, and f_i is one of n performance metrics. Variability and intrinsic parameters can be attached to the parameter space \mathcal{X} , integrated in the transfer function F , or taken into account as part of the constraints.

Pareto front

A set of parameters x_1 is said to dominate another set x_2 if the former fares at least as well in the performance space, and strictly better on at least one performance metric. In mathematical terms[BG15]:

$$f_i(x_1) \leq f_i(x_2) \forall i \in \{1, \dots, m\} \quad \text{and} \quad \exists j \in \{1, \dots, m\} | f_j(x_1) < f_j(x_2) \quad (5.2)$$

The standard design process can be improved by observing that the solutions of Equation 5.1.1 lead to Pareto-optimal configurations, where improving one performance can only be done at the cost of another: they are non-dominated, according to Equation 5.2. The set of all these optimal points is called the Pareto Front in the performance space (\mathcal{F}), whereas corresponding points in the parameter space (\mathcal{X}) make up the Pareto Set. The two sets are linked via the transfer function $F(x)$ that models the system, as illustrated in Figure 5.2.

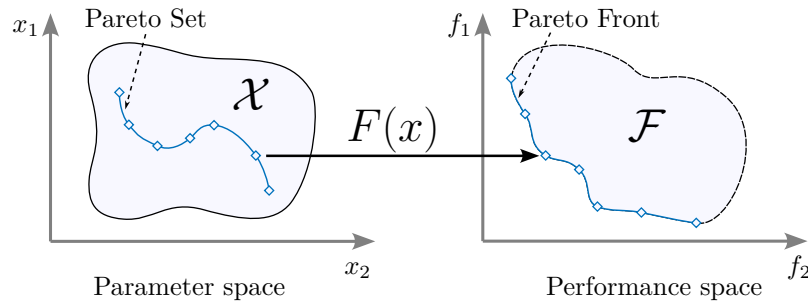


FIGURE 5.2: Parameter space \mathcal{X} and performance space \mathcal{F} , linked by model function F . The aim being to minimize performance metrics f_1 and f_2 , the Pareto Front is the set of optimal (non-dominated) values in \mathcal{F} , and the Pareto Set is the set of associated values in \mathcal{X} .

Pareto-optimal configurations correspond to matching sets of performance/parameter values $\{f, x\}$. Note that in case of non-conflicting objectives, the optimal solution can be the same for multiple performance metrics. In the extreme case where none of the objectives are conflicting, the Pareto set and front can be reduced to a single point[Deb01, p. 23].

5.1.2 Tool-assisted exploration

Since an exhaustive exploration of the parameter space is most often infeasible, and a manual selection of the interesting parameter values would be impractical due to the number of

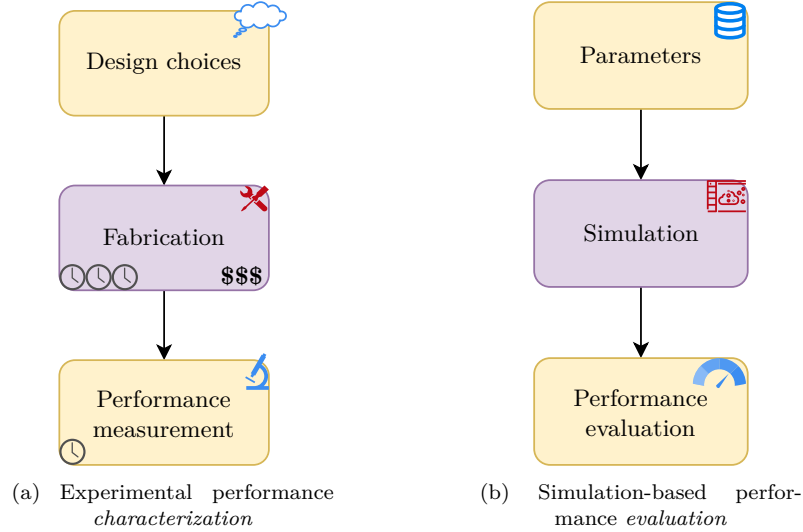


FIGURE 5.3: Experimental and Simulation-based approaches to performance measurements. Simulation approaches are faster and less capital-intensive, leading to much faster iteration cycles. However, they are less accurate.

simulation runs needed to identify these, automated **DSE** tools are used. Conceptually, these tools are quite simple: given a number of metrics to evaluate circuit performance, and a set of parameters that can be adjusted within specified bounds, the goal is to make the most out of available simulation time, by quickly converging towards Pareto optimums. This is done by using multi-objective optimization algorithms, and heuristics to select parameter populations.

Process

A circuit has to be specified as an optimization problem, clearly defining metrics that should be maximized and minimized, as well as constraints that make a design feasible or usable. A multi-objective optimization algorithm will then be applied to the problem, to select points of interest for the next set of simulations. Given the results of these simulations, the next batch of parameters can be identified.

The process is similar to a manual design iteration loop as illustrated in **Figure 5.3**: the designer selects parameters, evaluates the performance either through experimentation or simulation, and uses the result to refine parameters for the next experiment. Simulation-based workflows are easier to automate, and these iterations can be automatically performed until an optimal set of parameters has been found.

As illustrated in **Figure 5.4**, in the case of tool-assisted **DSE**, the designer is taken out of the design loop as it would be a bottleneck to fast iteration. The eventual goal of such a process is instead to provide the designer with points of interest in the parameter design space: multi-objective optimization algorithms should identify for each metric a set of optimal parameters, and allow the designer to pick the final compromise according to specifications. For instance, latency can be sacrificed in favor of energy efficiency. Parameters that result from the optimization process can be leveraged directly or guide design improvements. Possible sets of parameters would ideally be part of the Pareto set: they inform the designer that the compromise still maximizes every possible performance criteria.

Unfortunately, exhaustive exploration is required to prove that points are not dominated by other solutions. It is however possible to select the most optimal solutions encountered while exploring the design space, which maintains complexity tractable, while also providing both decent solutions and a representative view of the performance space [Dup22, p. 56]. This of course requires a multi-objective optimization algorithm that does not get stuck on local minima, explores the performance space efficiently, and that enough iterations were performed with it.

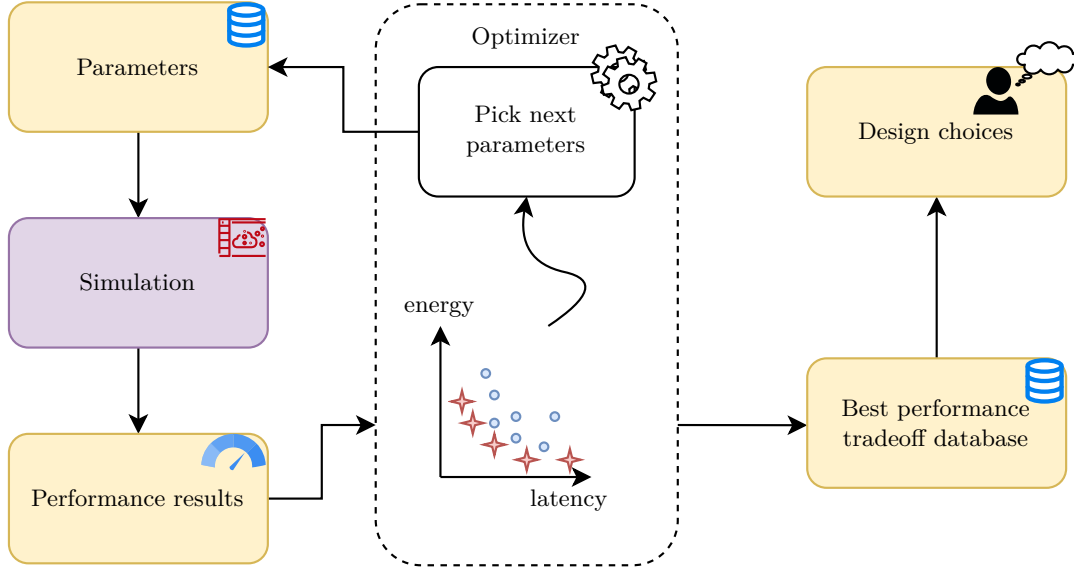


FIGURE 5.4: Automated DSE

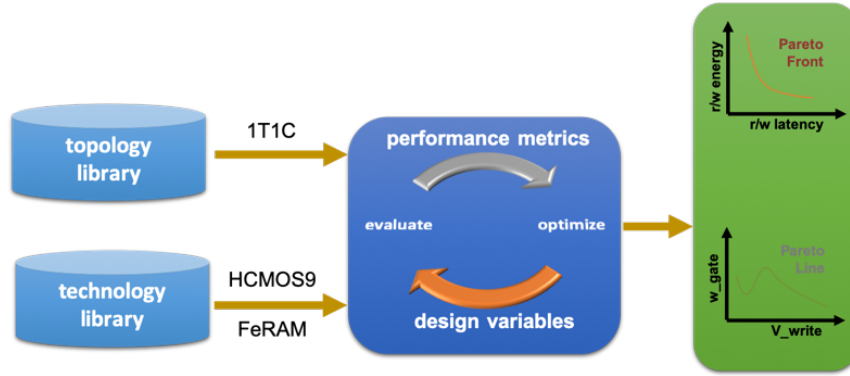


FIGURE 5.5: Pareto front and set generation flow

5.1.3 System-level benchmarking

Predicting the impact of newly designed circuits on system-level performance is a difficult exercise, as these circuits are only a small part of the whole system. At that level, performance is highly dependent on the circuit workload: different algorithms will have varied hardware circuits usage patterns. Moreover, new circuits enable radically different and novel architectures, that have to be compared against existing, better-known and better-optimized architectures.

To quantify gains that can be expected from various circuit designs and their possible applications, a flexible system-level benchmarking platform was developed, with the objective of bringing insights on the performance impact of both device-level and architectural design choices.

5.2 Design space exploration tools

5.2.1 LIFT optimizer

The structure for circuit-level Pareto-Front generation shown in Figure 5.5 has been implemented using an in-house optimization tool at ECL-INL[Fra12; Bri21]. Written in the MATLAB programming language, this tool code-named LIFT, leverages multiple optimization algorithms to produce a set of parameters from the previous simulations results.

The requirements are the following:

1. Specifying the parameters and their ranges
2. Specifying the metrics, whether to maximize or minimize them
3. Specifying operational constraints that allow validation of a circuit's functionality
4. Allow execution of a simulation and measurement of circuit performance.

Given the above conditions, two pain points were identified:

- System stability
- Communication with simulator (point 4 above)

The first point is a technical limitation of the exploration tool, as well as the designed circuits and models used. Given the range of possible parameters, some combinations cause malfunctioning simulations, triggering bugs in models, or convergence issues. Issues can be more subtle, with improbably high scores on some metrics, that then undermine the whole exploration process.

Generally, the process could be greatly improved with a database of previous simulations, and some input of the designer to allow specifying areas that should not be explored, while initializing the design space with previous values instead of completely restarting the exploration every time.

The second point is detailed in [subsection 5.2.2](#) below.

5.2.2 Cadence IPC

The design space exploration tool specifies the next set of parameters that should be considered for the design space exploration process. These parameters need to be communicated to the simulator. However, the **OCEAN®** simulator scripting language is designed around the opposite assumption, that it is the one controlling an optimization workflow. Therefore, an **Inter-Process Communication (IPC)** interface was devised to remotely control the simulator process.

As illustrated in [Figure 5.6](#), an **OCEAN** simulator process is first started and executes initialization scripts that will open file descriptors to wait for commands to be sent through.

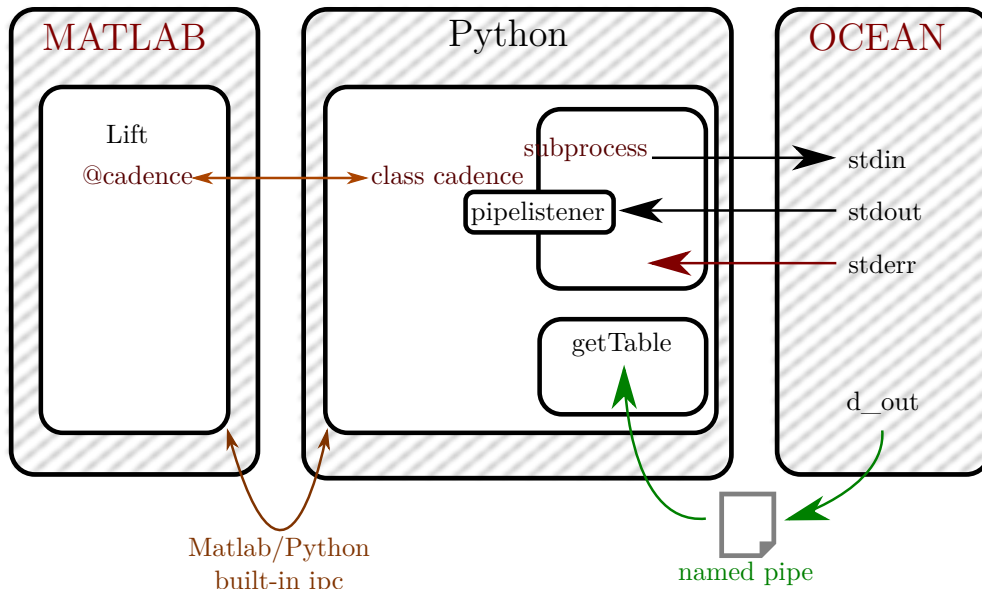


FIGURE 5.6: Cadence IPC architecture

The optimizer itself is implemented in **MATLAB**, which has limited capabilities for inter-process communication, such as no asynchronous file operations. This means that file descriptor opening and closing would deadlock the controlling process, so the **IPC** was rewritten in Python, which interoperates well with **MATLAB**'s foreign-function interface.

5.3 Design space exploration results

At the circuit-level, the objective is to generate data and mathematical models expressing trade-offs between performance metrics; and extract the associated design parameters. This data can then be used to inform system-level performance evaluation, as will be detailed in [section 5.4](#). In this section, **FeCap**-based 1T-1C cells are considered, as previously described in [section 3.2](#), as well as **FeFET**-based non-volatile logic gates from [section 4.4](#).

As a first approach, model cards representing typical performance metrics are extracted to feed into the system-level evaluation pipeline.

Results from automated **DSE** were disappointing, in part due to the instability of the ferroelectric model, that often resulted in simulator crashes, or in oscillatory outputs that confused the metrics extraction script (a situation colloquially known as “Garbage In; Garbage Out”), resulting in bogus performance metrics that completely undermined automated **DSE** algorithm performance. In fact, **DSE** algorithms are highly successful at identifying such problematic functioning points, which may make them useful for model development, testing and validation.

Pending development of improved models, it was decided to perform a manual **DSE** at first, exploring a grid of combinations along multiple dimensions. This provides an overview of the design space, and allows confirmation of general tendencies and influences that parameters have on performance metrics, but is much closer to the manual **DSE** process shown in [Figure 5.3b](#); making the extraction of Pareto sets a tedious process. Moreover, some points of the design space are redundant, and given the high-dimensionality of the problem, combinatorial explosion limits the exploration process to a few points per performance metric.

5.3.1 Sampling of 1T1C bitcell design space

Problem description and expected results

Memory performance is evaluated on three main criteria: memory capacity, energy consumption and speed. The first criterion is directly linked to the footprint of the bitcell and associated circuitry: the smaller the bitcell, the denser the memory and the larger the storage capacity (excluding **MLC** considerations). Energy consumption generally depends on circuit capacitance, and therefore should get lower as the ferroelectric capacitor area decreases. Likewise, as the capacitor area diminishes, charging time decreases, leading to increased operation speed and decreased latency.

These characteristics should therefore improve as the capacitor area is reduced. However, reducing the capacitor area lowers the observable difference between the readout of a logic high and logic low state (**memory window**), which imposes more constraints on peripheral circuitry, including the addressing and decoding mechanisms.

Besides capacitor area, it is possible to change the access transistor geometry: a wider transistor increases current, increasing both speed and energy consumption. A longer transistor decreases leakage current, as well as operation speed, and requires more energy to drive. Leakage current is not a meaningful metric in this case, as, unlike with **DRAM**, the capacitor does not need to retain its charge; the access transistor merely serves to open the circuit when accessing neighboring cells.

The impact of these parameters on memory window is more difficult to estimate. As detailed later in [section 5.3.1](#), it was chosen to integrate the **BL** current as a measure of the memory window, linking it to the **BL** energy. The predictions are summarized in [Table 5.1](#).

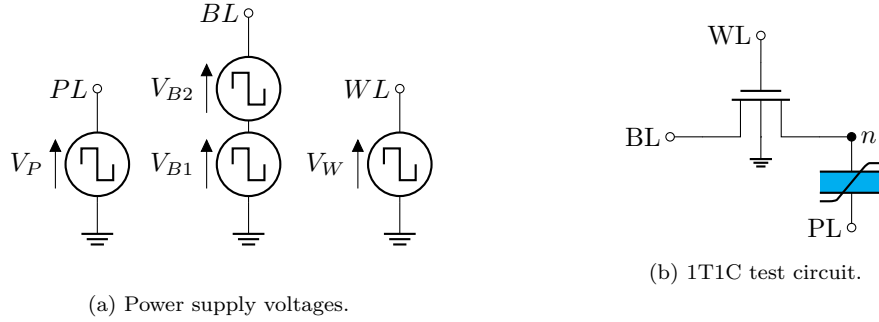
As demonstrated above, this problem is simple enough to anticipate most results, while having enough parameters for their interactions to be non-trivial. This is therefore a good first target to approach **DSE** for ferroelectric circuits, while generating results relevant for system-level performance evaluation.

Test circuit

The test circuit is a single bitcell as pictured in [Circuit 5.1b](#), operated as described previously in [subsection 3.2.1](#). This exploration made use of a Preisach **FeCap** model provided by NaMLab.

	Energy eff.	Speed	Density	Mem window
tW	—	++	—	=
tL	—	—	—	=
Ac	--	--	----	++

TABLE 5.1: Predicted performance impact of sweep parameters on 1T1C bitcell



CIRCUIT 5.1: 1T1C bitcell with voltage generators. Voltages supplied to the bitcell under test are **PL**, **WL**, **BL** and gnd.

Input voltages are generated by multiple periodic square signal generators, as shown in **Circuit 5.1a**. Operating at different frequencies, they make the input waveform completely periodic, which allows for cycling the ferroelectric before performing measurements. The resulting signals are plotted in **Figure 5.7**.

The chosen test pattern cycles the ferroelectric capacitor twice. Effectively, the test bench performs the following sequence of operations twice:

1. Write “0” to the ferroelectric
2. Read by writing “1” (switching current detected).
3. Write “1” to the ferroelectric
4. Read by writing “1” (no switching current)

The first cycle ensures the model is properly initialized, and makes the second cycle more representative of steady state operation. Performance measurements are performed on the second cycle. There are two series generators on **BL**, as this allows different characteristics for reading and programming pulses to be specified. The last pulse might be considered redundant, but it is representative of real-world operations, and provides a supplementary data point.

The parameters are indicated in **Table 5.2**. As visible there, voltage pulses are kept relatively long, and the output waveforms are observed to determine theoretical operation speed, instead of shortening them.

To better measure the ferroelectric material’s switching speed and status, the model is modified to expose the internal polarization **Pr** as a voltage source on an extra connector for the capacitor symbol, as visible on **Circuit B.3**. The output signal is shown in **Figure 5.7**, and allows observing the state of the ferroelectric oxide without relying on indirect indicators.

Problem definition

The bitcell geometry is fully parameterized as listed on tables **5.2** and **5.3**, to allow the impact of design choices to be studied. Technological parameters, typically **Pr** and paraelectric fraction, can also be parameterized to characterize the impact of variability on bitcell performance, though that was not the focus of this study.

The design space was sampled according to the script in **Listing A.3**, for every combination of dynamic parameter listed in **Table 5.4**: this exploration mainly focused on geometry, while keeping other parameters constant.

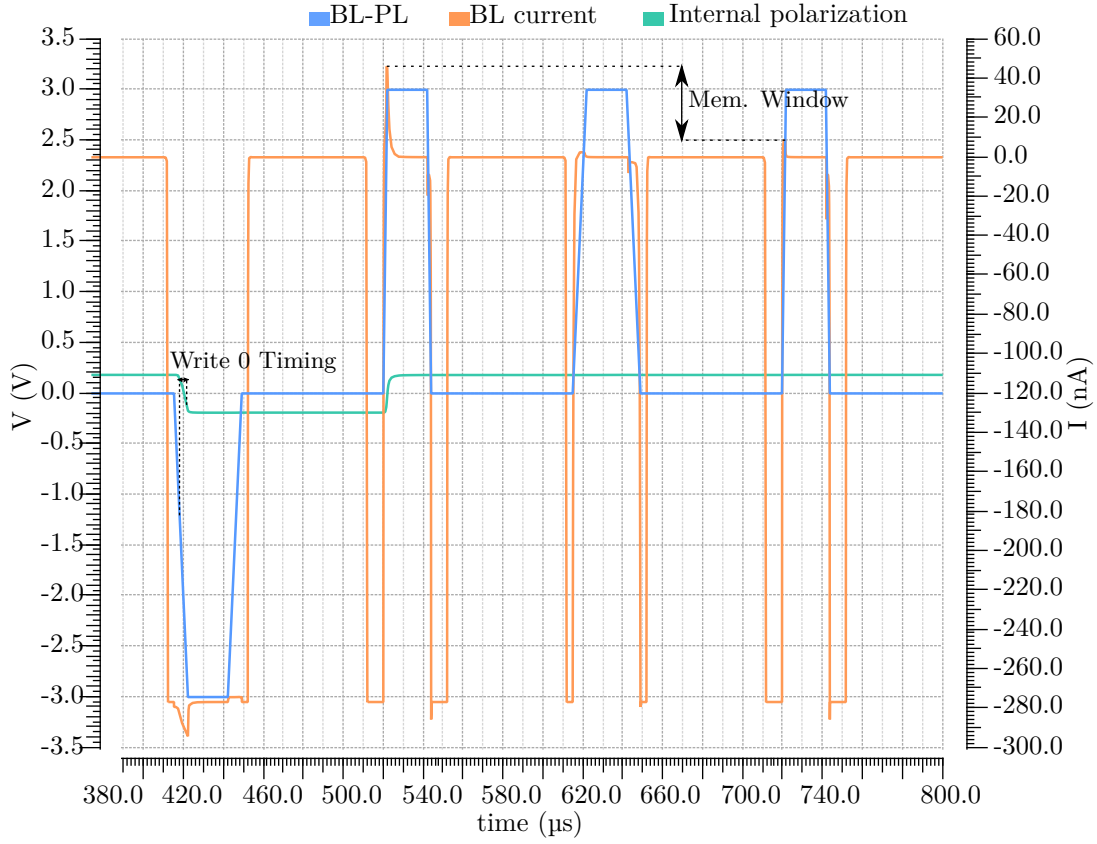


FIGURE 5.7: Test waveforms for 1T1C DSE. Internal polarization ● is monitored and timing is measured from the time the potential difference $BL - PL$ ● across the FeCap crosses V_C , up to when the internal polarization reaches 50% Pr. The peak current memory window is defined as the difference between peak BL current ● after reading a 0 or a 1. Note the current peaks corresponding to polarization reversal.

Parameter	VP	VB2	VB1	VW
Voltage1	0			
Voltage2	vprog		vread	vwl
Delay	20 μs	220 μs	120 μs	10 μs
Rise Time	triseprogPL	triseprogBL	trisereadBL	triseWL
Fall time	tfallprogPL	tfallprogBL	tfallreadBL	tfallWL
Pulse width	20 μs			40 μs
Period	400 μs		200 μs	100 μs

TABLE 5.2: Parameters for voltage generators. Some values are not specified numerically as they are used as parameters for simulation.

Transistor	
Length	L
Width	W
Capacitor	
Area	atot_fe
Paraelectric fraction	0.3
ε_r ferroelectric	30
ε_r dielectric	17
Tfe	10 ns
Psat	$2 \mu\text{C cm}^{-2}$
Pr	$19 \mu\text{C cm}^{-2}$
Vc	1.5 V
τ ferroelectric	10 ns
τ_f RC	50 ns
τ para	1 ns
Leakage resistance	$1 \text{ M}\Omega$

TABLE 5.3: Parameters for transistor and capacitor simulation. Numerical values are extracted from typical measurements for the technology, while geometry is parameterized to allow sweeping during simulation.

Dynamic parameters			Static parameters		
FeCap	Area (nm ²)	TL (nm)	TW (nm)	Slew rate	Voltages ^a Pr
	1.00×10^4	150	130	5 ns V^{-1}	3.6 V $0.19 \mu\text{C cm}^{-2}$
	2.25×10^4	170	140		
	3.24×10^4	200	160		
	9.00×10^4		180		

^aProgramming, read and word line voltages.

TABLE 5.4: Parameters for 1T1C bitcell DSE, both static (unchanged during every run) and dynamic. This DSE covers every possible combination of the dynamic parameter set, so $4 \times 4 \times 3 = 48$ data points.

Metrics extraction

Metrics are extracted from resulting simulation waveforms. More specifically, the script in [Listing A.1](#) measures the following quantities:

- Capacitor area (parameter)
- Access transistor width (parameter)
- Access transistor length
- Write time, as plotted in [Figure 5.7](#)
- Memory window, in quantity of charge
- Energy, write and read¹
- Peak currents, as shown in [Figure 5.7](#)

While [Figure 5.7](#) shows the peak current difference, another way to evaluate the memory window is to integrate the current over time, to extract the amount of charges transferred. The charge imbalance due to the repolarization current can be measured, and this value can be used to detect the stored value.

It is important to note that this is a “best-case” scenario, with the charge amount being compared to a reference readout of the same cell taken microseconds prior. Nevertheless, relative improvements to this memory window should be reflected in actual device operation.

A similar approach was taken to measure operation speed: the internal polarization is observed, and switching time is measured after the potential across the **FeCap** crosses V_C . More precisely, time is measured during polarization reversal from $V > V_C$ to $P > \frac{1}{2}Pr$. While a bad indicator of final memory speed, as power supply lines are driven with sources of non-zero impedance, this still provides an estimate of the theoretical speed limit of the memory bitcell. Obtaining more precise estimations of speed and power consumption would be difficult while simulating a single bitcell, and a precise approximation of these values is enough to provide an optimization objective and compare multiple versions of the same bitcell.

Results

Despite not being able to achieve completely automated **DSE**, an overview of the design space was obtained by manually selecting points of interest ahead of time, and validate multiple stages of the automated **DSE** pipeline. Exploration results were obtained and are plotted in [Figure 5.8](#), and the relationship between parameters and performance metrics is illustrated in [Table 5.5](#). Write time changes as expected, with it improving as transistor width increases and transistor length decreases: this suggests that writing speed is current-limited, or at least that higher currents allow faster writes, to an extent. At a capacitor area of $9.0 \times 10^4 \text{ nm}^2$ (corresponding to a capacitor diameter of 339 nm), write time is comprised between 88.9 ns and 117 ns, enabling theoretical operating frequencies of 8 MHz to 11 MHz. Interestingly, writing with a positive pulse a previously negatively-polarized **FeCap** leads to smaller simulated times of 49 ns to 59 ns. These times correspond to the switching speed of the **FeCap** itself. In practice, this value will be lowered by the address decoder capacitance, for both **WL** and **BL**, as well as the necessary sense amplifier and **ADC** circuitry. As expected, the values measured above are in the same order of magnitude as the τ_f RC model parameter indicated in [Table 5.3](#).

Measured memory windows range from 26.2 nC to 36.1 nC.

When contrasted with **DRAM** capabilities, where a sense amplifier is able to read a capacitor $C = 15 \text{ nF}$, which yields $Q = C \cdot V = 3 \text{ fC}$ at $V_{cc}/2 = 0.3 \text{ V}$, this suggest that there is enough margin to trade six orders of magnitude of memory window in favor of speed and surface, which will be investigated in future work.

Such a large memory window also invites the investigation of **MLC** memories, as capacitor area cannot be scaled below a size of about 100 nm of diameter due to ferroelectric grain size, as detailed in [section 2.1.1](#).

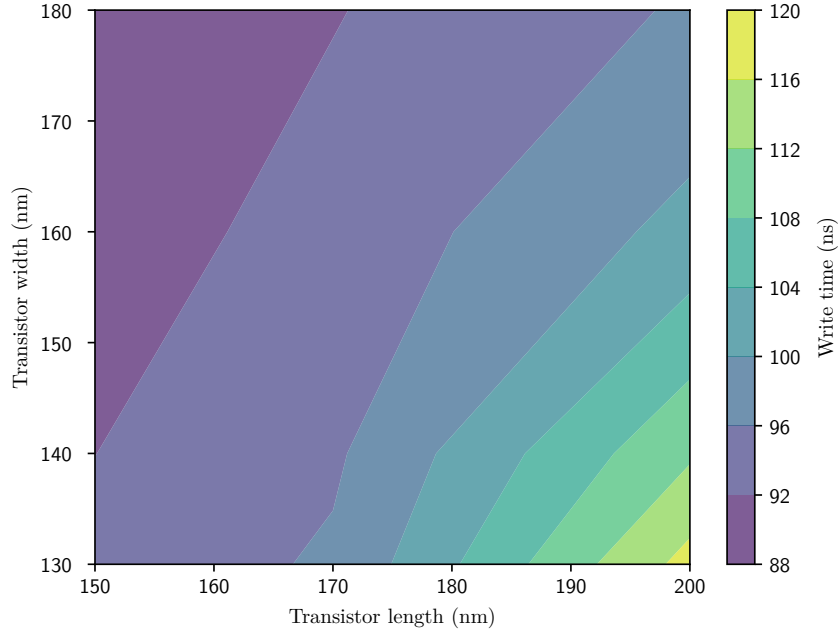
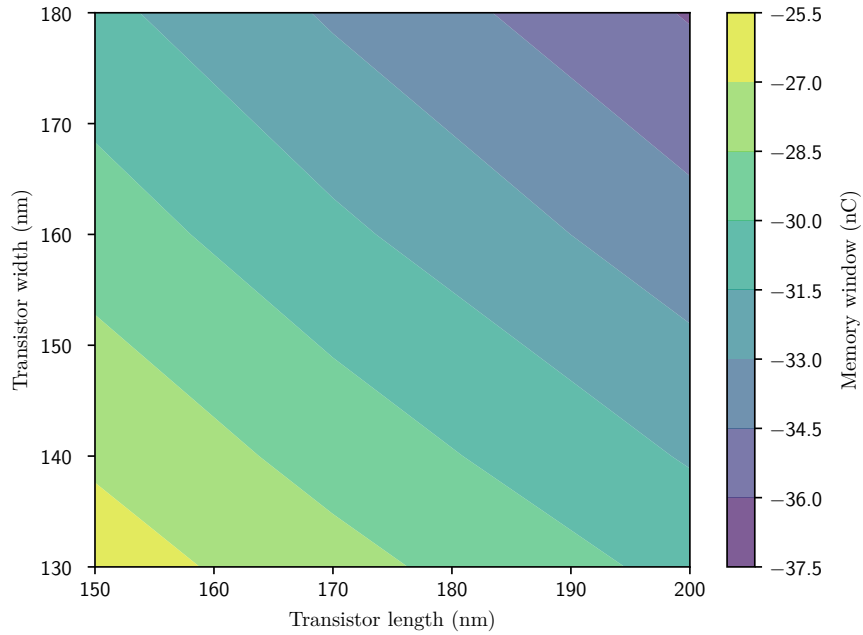
(a) Write time at capacitor area = $9 \times 10^4 \text{ nm}^2$ (b) Memory window at capacitor area = $9 \times 10^4 \text{ nm}^2$

FIGURE 5.8: DSE results for 1T1C exploration: time and energy vs transistor geometry. A clear improvement in write time can be seen as the transistor width increases, which allows the use of larger currents. Likewise, as transistor length increases, the current decreases.

The memory window is shown increasing with both width and length, and was found to be highly correlated (Pearson coefficient $\rho = 1$) with energy use. Smallest transistor dimensions are in the bottom-left corner, representing ideal values from an area utilization perspective.

Parameter	Energy eff.	Speed	Density	Mem window
transistor width	− ^a	+ ^b	− ^a	+ ^a
transistor length	− ^a	− ^c	− ^a	+ ^a
capacitor area	= ^d	= ^d	− − − ^e	= ^d

^aLinear: Pearson coefficient $\rho = \pm 1$; with similar slopes for transistor width and length

^bMore impactful as transistor length increases

^cLess impactful as transistor length increases

^dUnexpectedly: a large impact was instead expected

^eCapacitors being larger than access transistors, their area contribution is more important

TABLE 5.5: Extracted performance impact of sweep parameters on 1T1C bitcell, according to Figure 5.8 and Figure 5.9. When contrasting results with Table 5.1, the capacitor area impact was unexpected, as well as memory window results.

It is important to note that the above are only simulation results; they might be inaccurate.

One such possible inaccuracy under investigation is illustrated in Figure 5.9: a near-linear relationship between capacitor area and memory window as well as write timings would be expected, especially in light of the current-limited timings. In the same exploration, Figure 5.9 also shows that the FeCap area has almost no impact on these values. This suggests that other parameters have a much more pronounced impact, which was unexpected, and still remains to be investigated. The result may be realistic, or there could be an issue with either the model, or the simulation set-up, including with the metrics extraction script. Interestingly, as visible in Figure 5.8b, memory window was also found to be highly correlated with access transistor geometry, which may be linked to both the reduced performance impact of FeCap area, and the fact that memory window is determined from integrating the current that traverses the access transistor. The exploration will be re-attempted over a wider range of capacitor areas.

5.3.2 Non-volatile FeFET-based NAND gate (NV-NAND2)

Preliminary DSE was undertaken for a FeFET-based non-volatile NAND structure as described in subsection 4.4.1, using GlobalFoundries’s 28SLP technology and the same NaM-Lab-provided Preisach model used as an additional FeCap in the transistor gate stack to model a FeFET.

Test circuit and parameter space

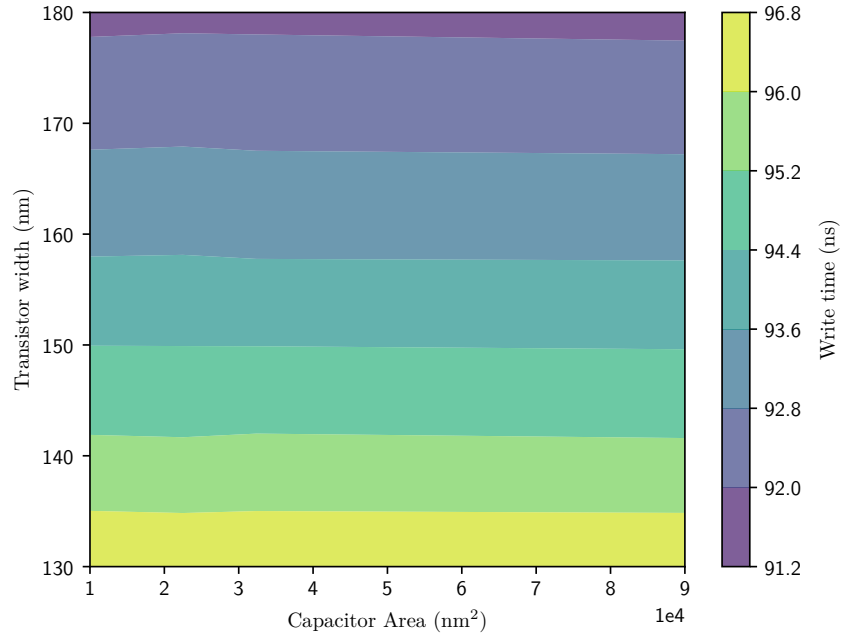
The design problem was set up according to parameters presented in Table 5.6, in a dynamic logic configuration, as shown in Circuit 5.2.

Design	NV-NAND		
CMOS technology	GlobalFoundries 28SLP		
Ferroelectric model	Transistor gate stack		
Design parameter		Minimum value	Maximum value
Logic/programming voltage	V	1.5 V/5 V	
Transistor width	tW	100 nm	1500 nm
Transistor length	tL	80 nm	1400 nm
Ferroelectric capacitance area	Atot	=tW*tL	

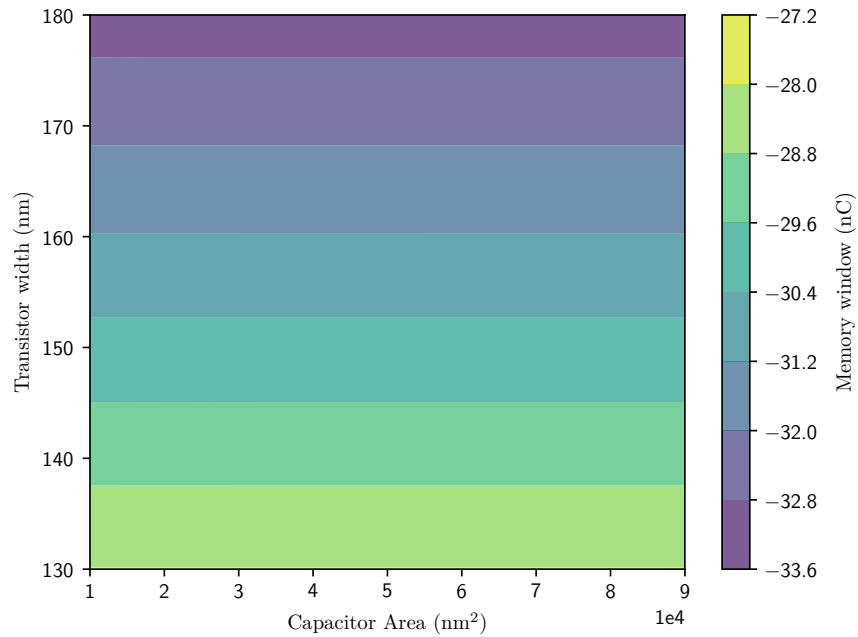
TABLE 5.6: DSE parameters for Non-volatile NAND gate performance exploration.

To reduce complexity of this first approach, parameters were kept to a minimum of two: FeFET width and length. Design space was first explored along a fixed grid, as shown in Figure 5.10a. The full code for this exploration is shown in Listing A.4, displaying how the IPC from subsection 5.2.2 is leveraged. Those preliminary design space exploration results are shown in Figure 5.10.

¹Energy to write a 0 over a 1 (ew₁₀), and both to read 0 and 1 states (er_{1_w0} and er_{1_w1} respectively) with a “write 1” pulse.

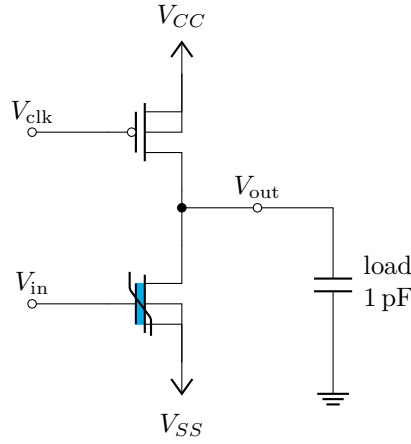


(a) Write time at transistor length = 170 nm



(b) Memory window at transistor length = 170 nm

FIGURE 5.9: DSE results for 1T1C exploration: time and energy vs capacitor area and transistor width (at constant length), showing almost flat relationship with considered area (*this result was later proved to stem from a problem with the experimental setup, see defense presentation*). Note that write time shown here are for repolarizing the ferroelectric with a negative pulse; simulation result show halved timings for positive pulses.



CIRCUIT 5.2: Dynamic two-input non-volatile **NAND** circuit (NV-NAND2) used for **DSE**. V_{SS} is set to ground (0 V) during normal operation, or to the programming voltage (chosen $V = 5$ V) for programming the “logic low” state. “Logic high” is programmed by raising V_{in} instead to the same programming voltage.

Performance space

Multiple criteria were examined:

- Operation of the logic gate: it is deemed operational if the output conforms to the logic table of a **NAND** gate, with the output voltage above V_{th} only when both inputs are low.
- “Precharge” energy: energy used for precharging the floating node, that will change depending on the geometry of the transistor, and dictates energy use of the logic gate.
- Voltage window: the exploitable voltage difference between an output logic high and output logic low level, for use in transistor-transistor logic.

The code used for performing metrics extraction is presented in **Listing A.2**, written in the **SKILL** programming language.

Results

Even with a reduced set of parameters and points, a few interesting patterns emerge from the data:

- Precharge energy seems to be more affected by transistor width; however, simulations indicate that energy costs are lowered as width increases, which is unexpected
- Voltage window seems to become more optimal at small **FeFET** areas; these metrics may be non-conflicting
- Smaller voltage windows seem to lead to invalid operations; which is expected

Since this exploration is quite restricted, it is possible that it misrepresents the distribution of the actual performance space. It is also important to note that the data displayed on these graphs is highly dependent on the accuracy of the metrics extraction script, which was not thoroughly investigated. Some results are interesting and require further analysis to understand. Such a simple problem already shows a relatively large dispersion of operating points in the metrics space; which lends credence to the **DSE** approach.

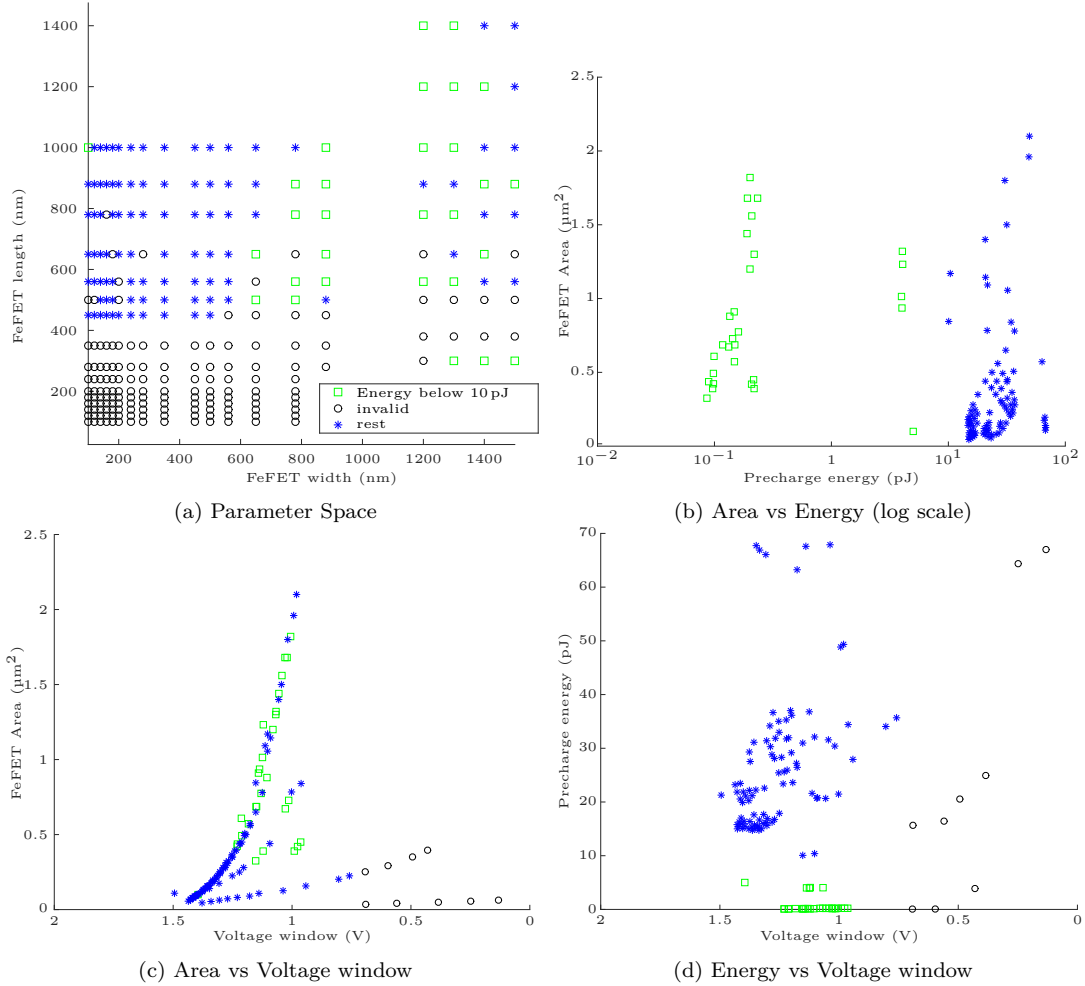


FIGURE 5.10: Preliminary DSE of the non-volatile FeFET-based NAND gate. 5.10a shows simulated points in the parameter space, while other figures show the corresponding points in the performance space. In every performance space plot, the optimal point is located in the bottom-left corner (small area, low energy and large voltage window). Green squares indicate the points that correspond to a precharge energy below 10 pJ. Black circles highlight the simulations that did not correspond to functional operation, as defined in section 5.3.2. 5.10c exhibits characteristics of non-conflicting objectives.

5.4 System-level benchmarking Platform

5.4.1 Introduction

Objectives

The main objective of this benchmarking platform is to evaluate the performance of a computing hardware architecture on a set of benchmarks, in order to:

1. evaluate the impact of circuit-level changes on system-level performance
2. compare its performance with other architectures on the same benchmark

Performance metrics can include throughput, latency, energy use.

This suggests a modular platform architecture, able to load a variety of computing architectures and benchmark definitions.

Use-cases

The aim of this tool is to guide design choices and to provide quantitative feedback on the gains expected from **normally-off** computing and coarse-grain **LiM**, while allowing an exploration of system parameters such as the memory duty cycle and power-down strategies, according to the application workload and memory access patterns. This feedback also enables co-optimization at the device level, with design parameters such as transistor footprints being guided from system-level performance considerations.

Acknowledgements

The design and implementation of this work was carried out with successive contributions by Master students Pierre-Etienne Polet, Luca Mozzone[Moz21] and postdoctoral researcher Marcello Traiola.

5.4.2 Scope of the benchmarking platform

Target System Architectures

The simulation platform aims to bridge the gap between device-level performance characteristics, and system-level performance metrics, while taking into account architectural particularities. As a result, it should be generic enough to accommodate multiple computing architectures[OCo+18]: **LiM** where small logic circuits are kept close to the memory, and **IMC** where the memory fabric itself performs computations. Some subsets can be distinguished among those paradigms: more generally, a distinction is made between coarse-grained and fine-grained **LiM**: while both associate logic and memory, the former uses dense memory arrays with peripheral logic for computation, and the latter leverages smaller memory cells inside the logic circuits, such as non-volatile filter coefficients, as illustrated in **Figure 5.11**. Therefore, the granularity refers to the size of the memory elements integrated with logic.

This exploration was focused on coarse-grain **LiM** as a first approach.

Extracted Metrics

Multiple memory technologies need to be evaluated: 1T-**FeFET**, 1T-1C **DRAM**-like structures with ferroelectric capacitors, as well as other conventional and emerging memories in order to provide a baseline comparison. Basic performance figures can be obtained from literature, simulation and experimental measurements, then fed into the platform as parameters for low-level memory cells.

In turn, the role of the platform is to quantify the energy cost of the operations, as well as achievable latencies, by tracking each cell's usage patterns. Simulated results are also compared to known theoretical outputs to track precision loss stemming from approximate computing circuits.

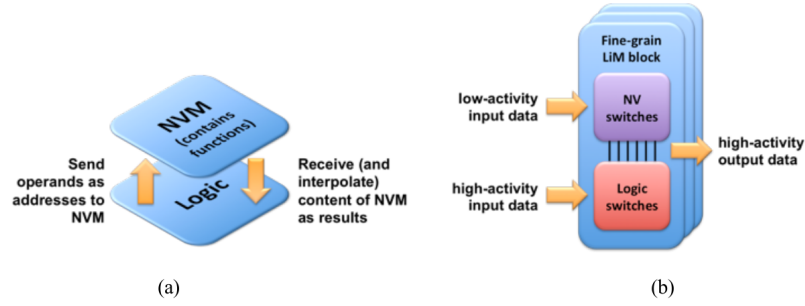


FIGURE 5.11: **LiM** concepts including non-volatile memory elements: (a) Coarse-grain approach using dense memory arrays with peripheral logic for computation , (b) fine-grain approach using small memory cells/elements within logic circuits.

Benchmarks

Arguably the most important part of an evaluation platform is the set of benchmarks that it can run. The objective is to obtain performance figures for realistic use-cases, so the platform should be able to interface with industry-standard benchmarks. As illustrated in [Figure 5.12](#), the current approach focuses on simulating the “enhanced memory” part of a computing architecture, connected to a **CPU** via a conventional address and data bus. As a coprocessor,

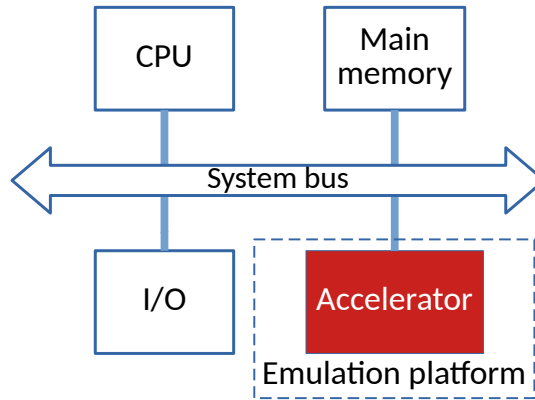


FIGURE 5.12: Positioning of a **LiM**-based accelerator on the system bus.

the memory can receive instructions through this bus, and benchmarks can therefore be evaluated as program execution traces replayed over the bus. This gives a lot of freedom on the generation side: the traces can be either handwritten synthetic benchmarks, an interfaced live program, an execution trace generated from a program, or extracted through compiler instrumentation[[Koo+18](#); [Mam+21](#)].

5.4.3 Implementation

The platform implementation methodology aims to achieve a framework in which **LiM** architectures’ performance can be estimated objectively and reliably, and subsequently compared. To allow such a degree of flexibility, simulations are performed at a high level to maintain them abstracted from the hardware, allowing different implementations of the same element in multiple technologies and with various architectures.

As a consequence, the design is modular in nature, which allows quickly and substituting of parts of it, and leaves room for future enhancements.

The simulator is developed with the **SystemC** libraries, that add hardware description facilities to the C++ programming language. This choice was made in part due to possible

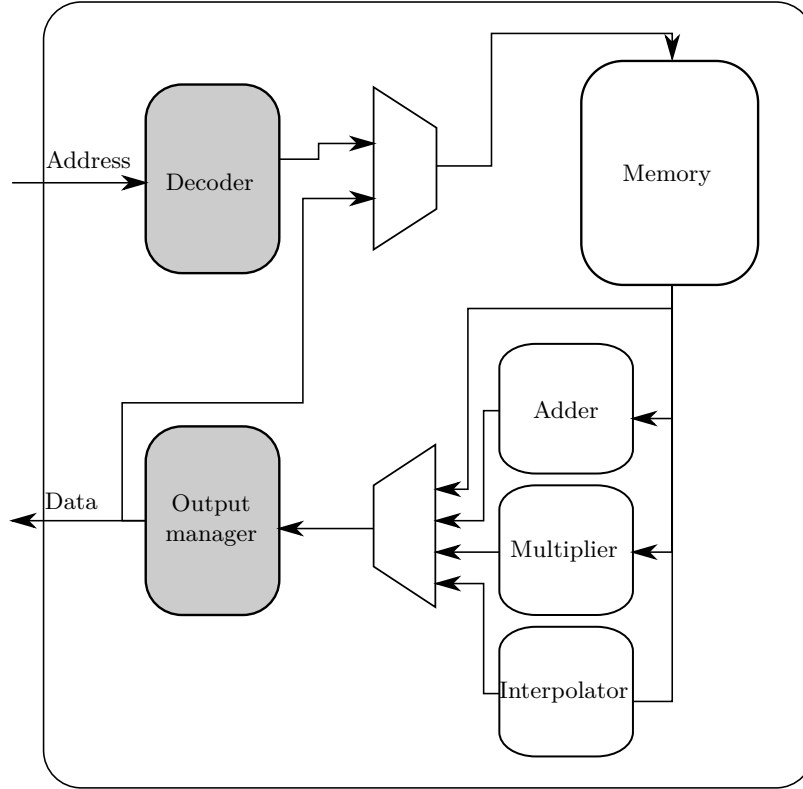


FIGURE 5.13: Diagram representation of the current platform architecture. Management modules are colored in gray, operation ones in white.

interoperability with high-level benchmarks through a C interface, as well as lower, **RTL**-level models of some components. This forces a modular approach to the design, which facilitates its evolution (i.e. the possible implementation of an **Arithmetic and Logic Unit (ALU)** for more accurate performance metrics and comparisons), helped by the object-oriented nature of the language. **SystemC** also offers facilities for recording signal history, which can be used to track performance metrics as well as the internal state.

Architecture

The architecture of the platform itself mimics that of a memory circuit, whose parts have been instrumented to reproduce, at larger scale, energy and latency metrics observed in the electrical simulation of smaller memory bitcell arrays. In this way, it is thus possible to extrapolate these performance metrics to the system level, using a suitable total amount of addressable memory. Specifically, this enables the comparison of realistic workload performance under normally-off, coarse-grain **LiM** or traditional memory (**DRAM**) architecture scenarios, where the simulated memory circuit is designed to operate under these different modes.

The internal structure diagram shown in **Figure 5.13** displays two different module types: management and operational. The first kind is not strictly related to parameter estimation, but manages control signals and reroutes data instead; the decoder and output manager are its main constituents. Operation modules are linked to hardware implementations: their description consists of model cards detailing their performance characteristics, extracted from simulations or the literature, and can be swapped or updated at run-time.

A distinction is also made between two different **LiM** usage patterns: **WB** and **Non Write Back (NWB)**, that differ in the way computation results are handled: the first one stores them back in memory for later use without sending them on the data bus, while the second one sends them back on the data bus for immediate processing by the **CPU**. As such, they are respectively akin to regular write and read operations from the data bus perspective. The

architecture reflects this in the output manager module, that allows it to operate in either mode.

Simulation pipeline

The platform was designed to take as input a series of instructions that can either be dynamically generated, or provided ahead-of-time in the form of an execution trace. As visible in Listing 5.3, this is very similar to regular processor opcodes, and consists of a series of memory accesses and LiM operations. These instructions are close to those a hardware implementation could use, although their encoding is not fixed. Their execution is simulated while performance indicators are updated. At any given time, the following quantities are available: overall latency of execution, total energy consumption and the error between the platform output and the expected result, caused by the finite – or approximate, depending on the architecture – precision arithmetic used for computation.

At the end of the execution, simulation traces are produced, reporting the evolution of the signals in time, along with the estimated energy consumption.

Another possible approach (that was not followed in this work) would be to sort instructions, and sum up their energy and latency costs. While the results would be similar in simple cases, and the architecture could be dramatically simplified as a result, this would not reflect more complex behaviors, such as dependencies between operations, and performance metrics that depend on state: for instance, it costs less energy to read a 0 from DRAM, or one of the two states for 1T-1C FeCap-based ferroelectric memories. Not taking this into account would preclude the comparative study of architectures optimized to take advantages of these asymmetries, such as a predictive scheme for reducing the need for WB when reading ferroelectric memory.

Decoder Control Module

This module parses an input trace file to simulate a data entry on the address bus, decodes the different input operations and generates control signals for other components. It is also responsible for ending the simulation at the end of the input file.

Output Manager Control Module

This control module selects where the data is headed. In the case of a NWB operation, the platform's output bus is driven, and the result is sent to the CPU. In the WB case, the write address is loaded from the decoder, then the output manager sends the result back to the memory for it to be written, avoiding the energy cost associated with transmitting data back to the CPU. This module also computes the error between the exact result as computed, and the approximated one which is being sent back on the bus. This is necessary, as an overflow can occur, or the operation could be implemented using approximate logic. The control module is also responsible for saturating the output when such an overflow is detected, if saturated arithmetic is desired.

Computation and performance tracking

The remaining modules in Figure 5.13 are the operational ones, that perform the actual computations: they enable the platform to simulate various operations, dictated by the memory word size, and maintain performance counters to track energy use and precision loss. They are described in more detail in section 5.4.4.

In order to measure the maximum achievable throughput and identify bottlenecks, the platform is fully asynchronous: operation modules communicate the estimated latency for each computation, which is tracked and summed up as a global performance counter, and dictates timings for the next operation.

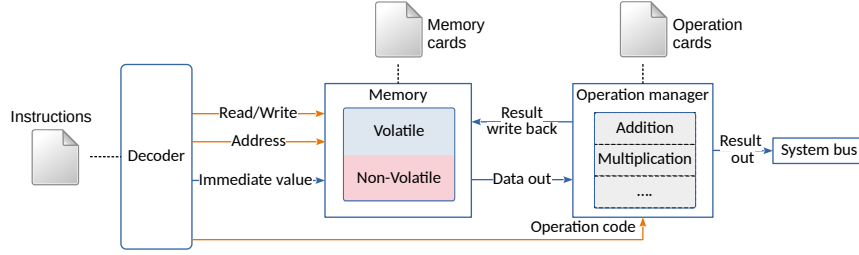


FIGURE 5.14: Simulation platform architecture, showing model cards, and the operation manager.

5.4.4 Operation modules and model cards

Memory module

While memory is considered an operation module in the platform's taxonomy, it is a special case, given how central its place is. It effectively sits at the core of the platform; providing an interface for other modules to read and write data to and from the memory matrix. Its behavioral model is very high level and generic, but can reproduce the behavior of multiple different technologies as encoded in different model cards, as illustrated in [Figure 5.14](#): either volatile memories such as **DRAMs**, or non-volatile ones like 1T-1C or 1T-**FeFET** Ferroelectric memories. The parameters listed on each card include read/write energy costs, latency, memory size, word length, refresh rate, as well as restart energy and latency.

Operation modules

As visible in [Figure 5.13](#), other operation modules are high-level implementations of different operations, with addition and multiplication currently available. Two multiplier variants have been designed: one with a standard hardware multiplier and a second one based on a sparse **lookup table (LUT)** combined with a bilinear interpolator. These modules rely on low-level performance parameters, such as energy consumption per computed bit and critical path latency, that are stored into model cards that can be loaded at run time, as shown in [Figure 5.14](#). This makes it possible to easily switch between different architectures and technologies for a given functional block, without changing its internal structure or the platform code itself. The values can be determined through experimental measurements, low-level simulations and from literature [[JGL16](#); [Vog10](#); [Sin+](#)].

Operation module implementation

The operation manager module includes a generic interface for an operation between two operands and a controller that realizes such operations, as shown in [Figure 5.15](#). This generic interface is implemented through the inheritance mechanism provided by the C++ programming language. Polymorphism enables us to easily add new operations, while enforcing their compliance with the interface: a generic parent operation class defines the virtual method `run`, that is then implemented by every operation.

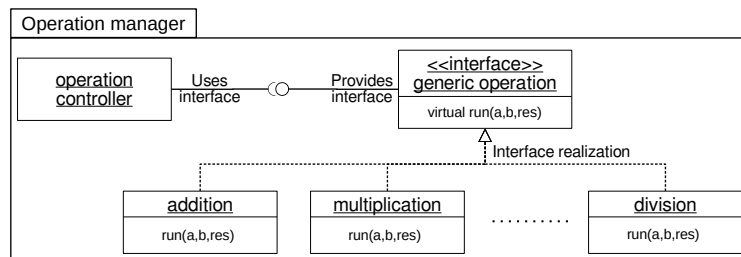


FIGURE 5.15: Illustration of the common operation interface

The decoder module receives the execution trace or accelerator opcodes as a list of operations to handle (instructions). It then interacts with the memory and operation controller

modules accordingly, as illustrated in [Figure 5.16](#). The operation controller maintains an array of pointers to each implemented operation, and calls the appropriate one (opcodes being currently array indices). This allows computation of the proper result, while the operation controller computes the delay and energy costs for the operation by reading the corresponding model card. Therefore, by providing distinct sets of model cards, different implementations of the same operation can be compared on the same benchmark, without recompiling the platform. Different memory and operation models can also be employed, though changing them either require a recompilation, or for models to be identified with different opcodes in execution traces. This allows comparing accuracy and results across different models.

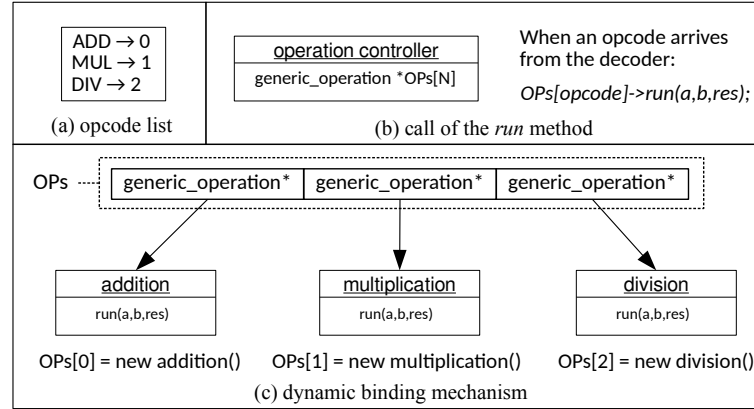


FIGURE 5.16: Execution diagram of the operation module

Model card structure

Correctly defining memory and operation module parameters (model cards) is key to achieve accurate performance estimations of the simulated architecture. [Listing 5.1](#) illustrates the structure of a memory model card, while [Listing 5.2](#) shows the corresponding operation model card structure.

LISTING 5.1: Memory model card template

```

1 v1      energy to read cell = 0 (pJ)
2 v2      energy to read cell = 1 (pJ)
3 v3      energy to write 0 over 0 (pJ)
4 v4      energy to write 0 over 1 (pJ)
5 v5      energy to write 1 over 0 (pJ)
6 v6      energy to write 1 over 1 (pJ)
7 v7      power needed to keep 0 (pW)
8 v8      power needed to keep 1 (pW)
9 v9      read latency (ns)
10 v10    write latency (ns)
11 v11    retention time (ns, 0 = inf)

```

LISTING 5.2: Operation model card template

```

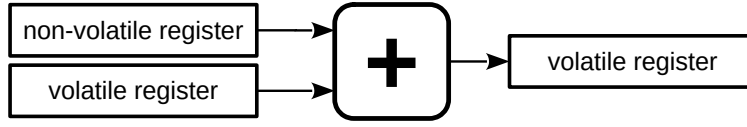
1 v1      number of bits of the operation
2 v2      energy per bit (pJ)
3 v3      latency (ns)

```

In both cases, the values (v1, v2, etc.) are read by the *Memory* and *Operation manager* modules and used in the simulation to keep track of the costs of the memory read/write and of the computations, respectively.

5.4.5 Example case: Adder

This simple example is meant to showcase the simulation platform capabilities, and illustrate its usage by modeling an accelerator that computes the sum of two values, as shown in

Figure 5.17. One of the input values is stored once in a non-volatile register and is used as**FIGURE 5.17:** Addition operation performed by the example accelerator

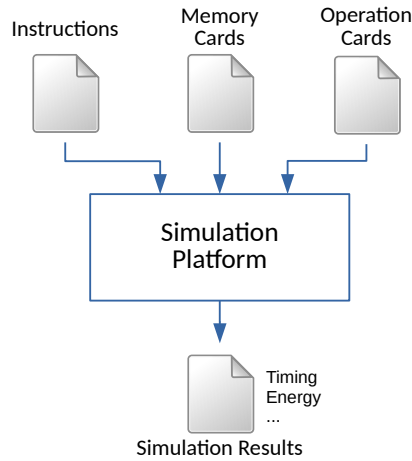
a constant. The other one is stored in a volatile register and changes over time. The result is stored back to a volatile register. For simplicity, let us suppose that the amount of energy and the latency to read/write one bit does not change depending on the previous/current values. The simple code that the decoder needs to implement this operation is the following:

LISTING 5.3: Execution trace for performing an addition

```

1  ww 0 64  #write volatile value 64 to the address 0
2  wnv 1 32 #write non-volatile value 32 to the address 1
3  ADD 0 1 2 #sum values at addresses 0 and 1 and write result at address 2
  
```

This code makes use of two registers (memory addresses), one in volatile memory, and one in non-volatile memory. The code implementing the operation is then called with the ADD instruction, which reads parameters such as latency and energy cost for the operation from the corresponding operation model card. As this instruction specifies a register to write the operation result value to, the output manager operates in the **WB** mode, and the associated energy cost is tracked.

**FIGURE 5.18:** Inputs and outputs of the benchmarking platform, showing model cards as input, and performance evaluation as output

As shown in **Figure 5.18**, the platform can simulate a combination of execution traces, memory and computation model cards, and extract corresponding performance estimation. **Table 5.7** lists the simulation results in terms of energy and delay for the addition operation performed with adders of different input bit widths (8, 16, and 32), as well as different implementations (expressed as energy and delay needed to perform the addition).

Simulations were realized by simply changing the memory and operation model cards according to the parameters in the table, between each simulation run, without recompiling the platform executable. This demonstrates the flexibility of the chosen architecture, which was designed to be integrated in a **DTCO** loop (as detailed in **section 5.6.2**), and thus requires minimal friction for changing parameters.

5.5 System-level exploration results

As a way to provide feedback and validation during the platform development, simple operation modules and benchmarks were prepared in order to demonstrate the ability to provide

Parameter				Unit
Adder bit-width	8	16	32	bit
NV read	0.01	0.01	0.01	Energy (pJ/bit)
NV write	0.05	0.05	0.05	
Volatile read	0.1	0.1	0.1	
Volatile write	0.2	0.2	0.2	
NV Addition	0.5	0.5	0.5	
NV read	0.05	0.05	0.05	Delay (ns)
NV write	0.1	0.1	0.1	
Volatile read	0.1	0.1	0.1	
Volatile write	0.15	0.15	0.15	
NV Addition	8	16	32	
Total energy	8.48	16.96	33.92	pJ
Total delay	8.55	16.56	32.56	ns

TABLE 5.7: Parameters and simulation results for the adder example. This table shows three sets of parameters, with different adder bit-widths, as well as the simulated total energy and delay per operation. Parameters were provided in model cards and changed between each simulation, without recompiling the simulation platform executable.

estimations and feedback on both architectures and technologies. The tool can be used to detect critical parameters during the development of LiM devices, and to acquire a better understanding of how they affect the whole system performance. For the simulation results that follow, module parameters in Table 5.8 have been derived from the literature[JGL16; Vog10; Sin+], as well as estimated values. As such, these should be taken as a demonstration of the analysis capabilities of the platform, rather than at face value.

Parameter	Value	Unit	Source
energy to read ‘0’	1	nJ	Simulation
energy to read ‘1’	2.5		
energy to write ‘0’ over ‘0’	0.5		
energy to write ‘0’ over ‘1’	2		
energy to write ‘1’ over ‘0’	2		
energy to write ‘1’ over ‘1’	0.5		
power to store ‘0’	0.1	mW	Estimated
power to store ‘1’	0.1	mW	
restart energy	5	nJ	
shutdown latency	200	ns	
read latency	20	ns	Experimental
write latency	20		

TABLE 5.8: Energy consumption parameters as used for simulation in the ferroelectric 1T-1C memory cell model card. Parameters were derived from simulation where possible, otherwise obtained from literature or estimated.

5.5.1 Normally-off use-cases

In cases where memory accesses or computation are infrequent, it may be beneficial to power the system off, even when that entails extra work, thus extra energy consumption, for saving and restoring the state to and from Non-Volatile Memory[Win+20]. Energy efficiency gains are instead expected to be made by lowering the static power consumption of the system. In such cases, it is of primary importance to understand the tradeoffs between shutdown energy

overhead and static energy consumption, and in particular quantify the limiting (maximum) activity duty cycle below which energy can be effectively saved through system shutdown.

The benchmarking platform can also be leveraged to estimate how energy usage varies from one memory technology to another.

Figure 5.19a shows how the benchmarking platform can be leveraged to produce energy consumption plots from a collection of benchmarks and model cards. Figure 5.19b and Figure 5.19c shows respectively the energy and power use during a matrix multiplication benchmark in two scenarios: one with shutdown, and the second without. As a demonstration of the platform capabilities, this benchmark used arbitrary energy consumption figures. Nevertheless, gains achievable with **normally-off** computing are clearly visible on the resulting graphs.

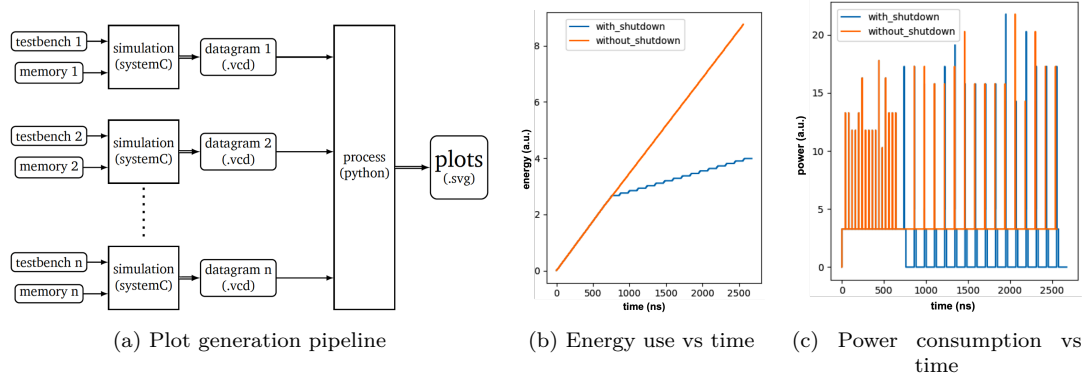


FIGURE 5.19: Normally-off simulation showing cumulative energy consumption during 16 write and 16 read operations with and without shutdown strategies, next to an illustration of the pipeline used to generate the graphs.

This benchmark was kept as a baseline capability demonstrator as the platform evolved.

5.5.2 Interpolator simulations

An interpolator is one possible **LiM** implementation, whereby output data of an n -input function is stored in an n -dimensional array (in a **LUT**-like fashion). As storing every combination is not feasible as the number of dimension and states grow, a sparse array is used instead, storing only a few data points, and the result is interpolated from the available points. With this approach, area and performance requirements can be traded between logic and memory elements. Depending on the interpolation function (for instance, linear or bilinear) and the array density, the result can be more or less accurate, allowing to sacrifice accuracy for complexity or performance gains.

The simulation objective is to investigate the competitiveness of a **LUT** and interpolator-based multiplier architecture with traditional logic. The interpolator function was chosen to produce non-approximated results, as a baseline comparison point: it performs a bilinear interpolation of the precomputed values[Moz21, p. 49]. It has been subjected to extensive tests to understand how its performance varies with respect to physical and technological parameters: it was tested with both pseudo-random and the exhaustive list of 8-bits signed integer input operands, while measuring the average energy consumption per operation against the **LUT** size.

Figure 5.20 plots compute energy versus normalized **LUT** circuit area, which is equivalent to the memory size in bits:

$$A = \frac{A_{\text{Array}}}{A_{\text{bitcell}}}$$

The same analysis was conducted with both current memory technology parameters for the memory model card, and with improved performance values, so as to identify the required memory characteristics for this architecture to be competitive. These results show that the interpolator-based architecture's energy efficiency cannot be competitive with regular multipliers using current memory technology (as detailed in Table 5.8), regardless of the memory array size. Only a 80% reduction in read energy inverts the tendency, causing computation

energy to decrease as the array size increases (with the interpolator having less computations to perform). On this improved memory technology node, the increased power consumption of larger memory arrays is small enough to be offset by the energy saved by reducing the compute load of the interpolator. This would result in an architecture that performs a multiplication more efficiently than with logic-only approaches starting at a 1024 word **LUT** size, where simulated energy consumption becomes lower than without memory. Beyond this value, **LUT** size choice becomes an Energy-Area trade-off.

Depending on advances needed to reach that improvement, interpolator would also need to be competitive against improved logic nodes in that case, which makes it unlikely to be competitive in practice, unless memory technology advances faster than logic (which was historically not the case), or if the chosen application precludes the use of advanced logic nodes (which can be true for memory designs). **FeFET**-based memories, by virtue of not requiring a **WB** mechanism, may be better suited to this use-case, as **LUT** only changes when the performed function changes. Another avenue for performance improvement is to lower the complexity of the interpolator by decreasing the accuracy of the results, making use of the approximate computing paradigm. While evaluating this operating mode was a design goal of the evaluation platform, this has not been investigated.

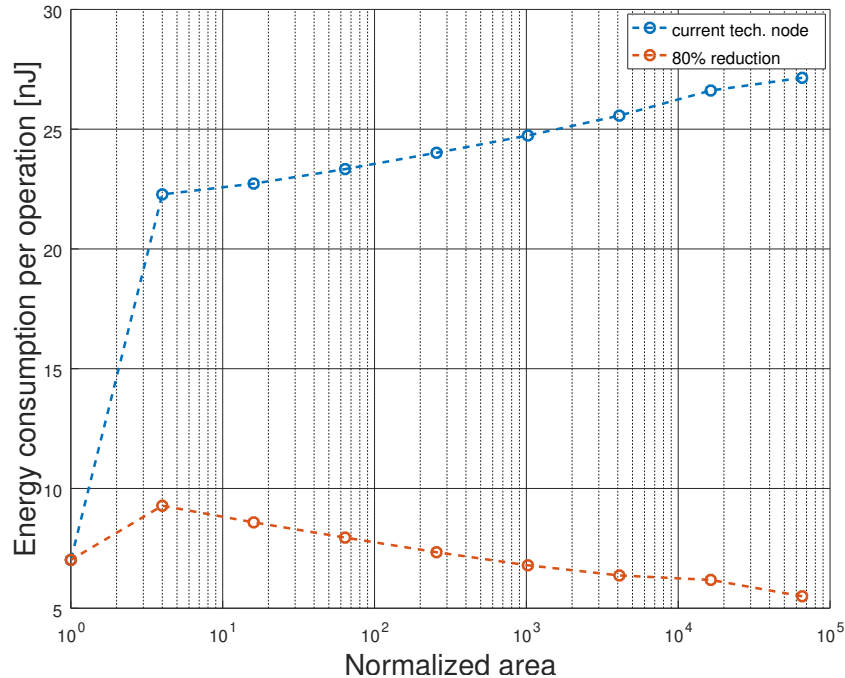


FIGURE 5.20: Average interpolator energy consumption per operation versus **LUT** normalized area. The first value considers the interpolator without **LUT** ($A = 0$). The plot shows possible energy efficiency gains over regular multipliers, if the energetic cost of reading **Non-Volatile Memory** can be improved at least 80%.

5.5.3 Matrix multiplication benchmark

As a slightly less synthetic benchmark, and as a stepping stone to more complex algorithms, the behavior of a matrix multiplication was investigated. This benchmark consists in multiplying two 8-bit signed integer 5×5 matrices using a naive matrix multiplication algorithm. Figure 5.21 displays the total energy consumption of the **LiM** accelerator against word size used for storing and transmitting the matrix multiplication result, for both **WB** and **NWB** cases. A small but growing energy difference can be observed between the two scenarios, due to the contribution of the output bus sending computation results to the central processing unit. This cost is always present in the **NWB** case, while it is absent from the **WB** case. It is also important to note that, as a quirk of the precision evaluation mechanism, data is

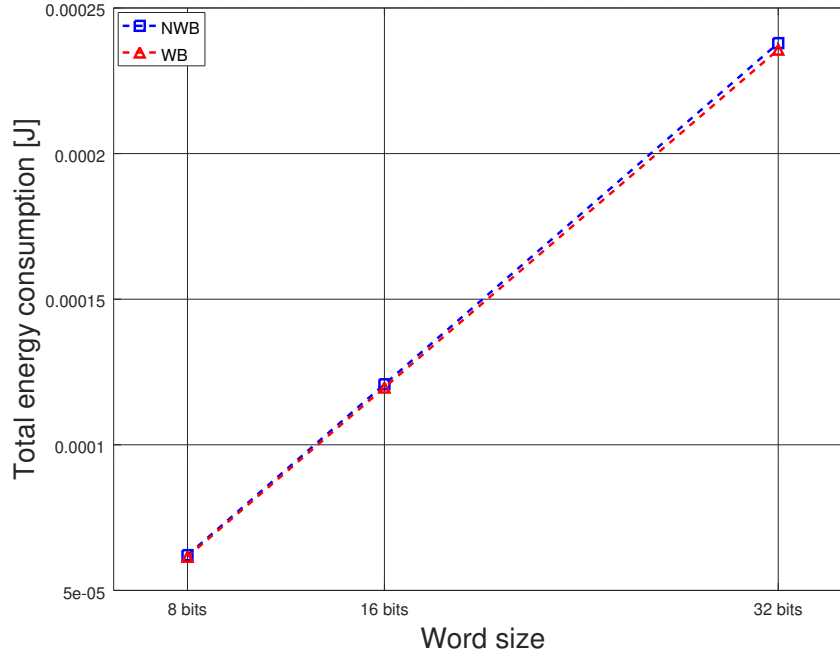


FIGURE 5.21: Total energy consumption versus word size for matrix multiplication benchmark.

currently saved to memory in the **NWB** case, which unfortunately contributes to the energy use. This benchmark highlighted the need for more precise evaluation mechanisms to evaluate the power consumption of the communication bus and central processing unit, which are currently estimates.

5.6 Conclusion

5.6.1 Design-Space Exploration

In this chapter, a fully automated **DSE** pipeline was presented. While model instabilities prevented its prolonged use as an autonomous optimization platform, the modular architecture allowed us to manually screen the design space without using an automated feedback loop prone to converge into edge cases. This revealed potential methodology improvements, as well as unexpected findings that remain to be investigated in-depth, such as the low impact of capacitor area on memory window². Perhaps more importantly, multiple components such as the **IPC** described in subsection 5.2.2 have already proven³ valuable in other projects[Poi22, p. 103].

Another product of this design space exploration was the partial mapping of the resulting performance space for ferroelectric-based circuits. While these results have not been evaluated against experimental data, they are a reasonable starting point to begin evaluating the system-level performance impact of these new circuits.

Model instability and accuracy

As a simulation-based approach, automated **DSE** only reflects how compact models interact at different points in the design space. It is therefore crucial for these models to accurately represent the performance of devices and circuits. Model stability issues prevented the exploration of some design space areas. Moreover, instabilities were often accompanied by inaccuracies such as high-amplitude oscillations due to convergence issues, which prevented fully automating performance metrics extraction. The lack of detailed experimental data for

²This result has since been shown to stem from an issue with the setup, refer to the defense presentation.

³Most of this work was also reused by replacing **LIFT** with **pymoo**[BD20].

circuits based on ferroelectric devices also prevented evaluating the model accuracy, lowering the confidence in results presented in this chapter.

While progress was made on automated DSE, it highlighted a need for improved models. As experience and experimental data on HfZrO_2 -based ferroelectric circuits and devices is accumulated, better models should enable deeper explorations. Simpler models such as the one presented in Listing 2.1 may provide interesting insights, trading accuracy for simulation performance. Other approaches to modeling may also reflect more complex ferroelectric behaviors[Den+20].

5.6.2 System-level benchmarking platform

Current status

Work was conducted towards the realization of a system-level benchmarking platform, taking as parameters performance metrics from simulations operated at the device- and circuit-level. The main evaluation tool was realized in SystemC, which enables direct interfacing with existing industry-standard benchmarks for applications such as image recognition.

While the platform only began producing results, it shows promising potential as a tool to explore further architectural variations, as well as the system-level impact of device-level performance improvements.

It may also become a useful tool to develop and benchmark LiM-aware compilers. However, competitive tools such as Gem5-X[Qur+21] have since emerged, which it should be re-evaluated against.

Further refinements and analysis

Current results only show that minor improvements may be achieved with the evaluated architecture. This is in part due to an incomplete architecture and benchmark library. Moreover, the uncertainty of some model card parameters is relatively high due to DSE results, and conservative estimates were used in multiple places, as the realization of the benchmarking platform was the focus. Consequently, these results show interesting trends and areas to pursue further, but absolute performance values are highly uncertain.

Part of the issue, is that the platform currently only considers energy and latency contributions of LiM coprocessor architectures, without comparing them with CPU-based computations. Extending the platform to cover that part of the system is likely necessary for a complete breakdown of the energy use at the system level: while the current architecture is able to contrast multiple coprocessor implementations, the comparison may be incomplete without including CPU figures.

Design-Technology Co-Optimization

Circuit-level Pareto-Fronts generated from DSE as described in section 5.1 help the designer choose performance trade-offs adequate with the design goals. However, estimating system-level performance on realistic benchmarks is the objective of the benchmarking platform, and cannot be done accurately at the circuit level. Therefore, the ultimate objective is to approach the problem in a DTCO way, combining DSE with the system-level benchmarking platform for performance evaluation.

Figure 5.22 shows the envisioned architecture, with the right part performing the optimization loop described in section 5.1 with the tools from section 5.2, extracting Pareto fronts from performance results obtained with the system-level platform visible on the left side.

Unfortunately, this was not explored further due to a time constraints, although multiple milestones were passed on the way to this ultimate objective. Indeed, to make such an optimization loop possible, every single piece has to be working perfectly, from the DSE pipeline (including models) to the system-level performance evaluation platform. This multi-stage approach to DTCO yielded modular and generic tools, allowing their reuse in other projects.

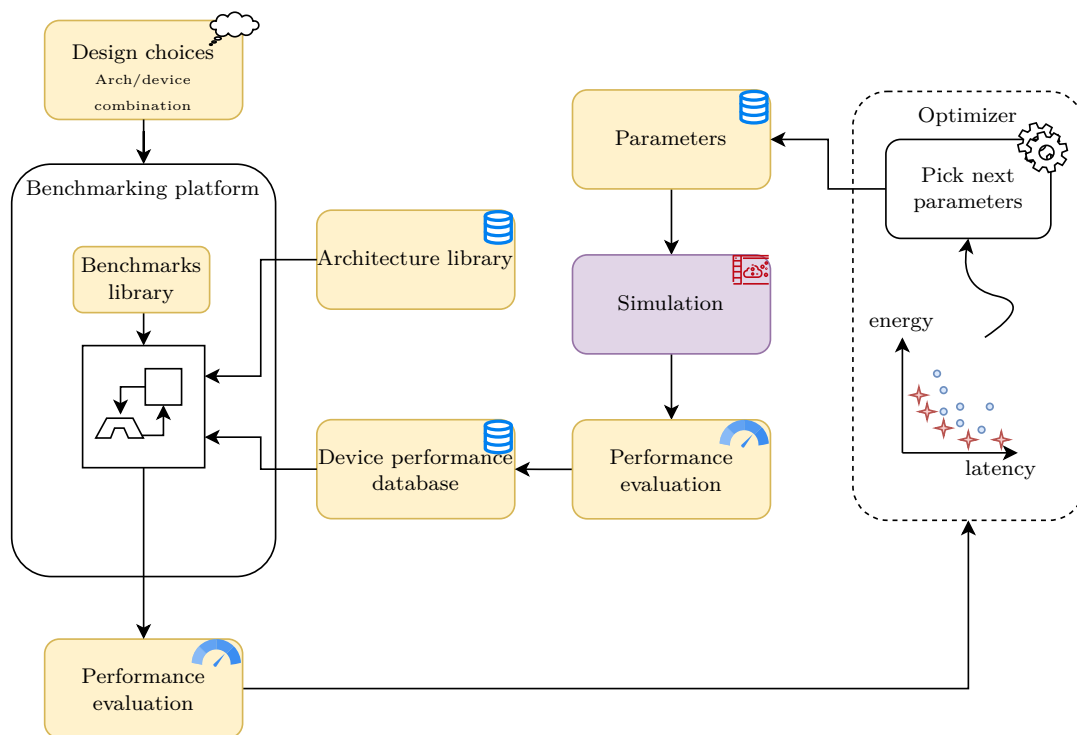


FIGURE 5.22: Complete automated **DTCO** pipeline as envisioned, combining system-level performance evaluation, left, with device-level parameters selection and **DSE**, right.

Chapter 6

Conclusion

Contents

6.1 Back-End of Line ferroelectric technology	143
6.2 Current FeFET strengths and limitations	144
6.2.1 Future of FeFET technology	144
6.3 Automated DSE and modeling	144
6.3.1 Modeling issues	144
6.4 System-level performance evaluation	145
6.5 Short-term perspectives	145
6.5.1 Remaining characterization work	145
6.5.2 Future simulations	145
6.6 Future considerations for ferroelectric technology	146
6.6.1 Space efficiency	146
6.6.2 Control signals	146

6.1 Back-End of Line ferroelectric technology

In [chapter 3](#), BEoL ferroelectric capacitors were discussed. Thanks to their relatively low annealing temperature of about 450 °C for 30 s [[Bou20](#), p. 44], ferroelectric transistors can be deposited above metal layers without damaging previous structures. This lowers precision requirements when depositing [FeCaps](#), as both top layers and capacitor structures generally have relatively large footprints, and decouples capacitor size from transistor sizes.

When contrasted with [FeFETs](#), BEoL technology gives more freedom to the designer, by allowing both a lot more flexibility with capacitance matching, and a bigger variety of circuit architectures. Indeed, while [FeFET](#) functionality can be replicated using discrete [FeCaps](#) (resulting in [PsFeFET](#) or [FeMFET](#)), including BEoL ones as detailed in [section 3.3](#), the opposite is impractical [[Sle+19b](#)].

While results for a more in-depth comparison of [PsFeFET](#) and [FeFET](#) are not available yet, preliminary results are promising and validate normal [FeFET](#) operation. Experimental results show that the [PsFeFET](#) cell is able to memorize and recall data, with a relatively important memory window, that could be further subdivided into multiple levels for [MLC](#) operation. 2T1C characterization results also confirm these findings, with the only difference being the use of a separate transistor to access the floating node, thereby lowering the required operating voltages and simplifying the programming sequence. It will be interesting to evaluate leakage current and retention of these structures in more detail, although for memory applications the stored value can always be read destructively in order to refresh the memory as [DRAM](#) would be.

A destructive [TCAM](#) cell was also designed and fabricated. While more recent, non-destructive designs [[Yin+19](#)] make this one partially obsolete, it may still be useful for [LiM](#) applications. However, as most of the functionality can be replicated with the 2TnC cell, characterization efforts will first be focused on the latter.

6.2 Current **FeFET** strengths and limitations

FeFETs display appealing features for hardware implementations: non-volatility with a retention time of up to 10 years (extrapolated), **CMOS** compatibility, known array write schemes, moderate write voltages, solder reflow stability as well as fast read and write operation. Moreover, they naturally combine logic (**MOS** transistors) and memory (ferroelectric oxide) elements.

On the other hand, they face charge trapping effects, that prevent read-after-write operations[Kle+21]. Trapped charges can also cause endurance issues, screen the memory window and change the V_{th} , especially in small devices. Strong electrical fields damage the transistor gate oxide, limiting the cycling endurance to around 10^4 to 10^6 cycles. Thicker gate stacks necessitate higher voltages, complexifying their integration with sensitive **CMOS** transistors.

Therefore, the integration of **FeFETs** into novel **LiM** applications makes most sense where an internal value, saved in the ferroelectric layer of the **FeFET**, is used to process an externally provided data stream. They are best used when infrequently repolarized, where direct read after write is not necessary, and where fast read and write operations are needed.

6.2.1 Future of **FeFET** technology

According to the presented analysis, **FeFET** are relatively unwieldy to use, requiring higher voltages and large transistors, as well as a dedicated programming circuit with the associated addressing mechanism. This can be greatly improved by using **FeFETs** constructed with **BEoL FeCaps**, as described in section 3.3 under the **PsFeFET** name, even more so as decoupling the capacitor from the transistor allows inserting an extra access transistor, seen in section 3.5. This access transistor can be integrated in the addressing mechanism to program the ferroelectric oxide, lowering the programming voltages and their nefarious effects on the transistor gate oxide. Moreover, decoupling both allows sharing a single capacitor between multiple transistors, which as described in subsection 4.3.1 can be especially beneficial in case of **CMOS**-based **FeFET** circuits, but can also unlock other finer control and other **LiM** operations, as illustrated with the 2TnC circuit presented in section 3.5.1.

PsFeFET and 2T1C could therefore possibly become more dominant in the future, unless their retention characteristics are incompatible with the application. Moreover, some applications described in chapter 4 can also be implemented using regular capacitors in series with gate transistors, in a **FGMOS**-like structure.

6.3 Automated **DSE** and modeling

Automated **Design-Space Exploration** on ferroelectric circuits was quickly limited by the maturity of available models. Nevertheless, manual exploration enabled the development of individual pieces of the automated pipeline, and highlighted interesting areas to explore, such as the lower-than-expected effect of capacitor area on memory window detailed in section 5.3.1. These results may be simulation artifacts, stemming from the interaction between different stages of the pipelines, or from the models themselves. This is being investigated, and will be compared to experimental results when available.

Tools produced for the automation pipeline, among which an open-sourced **IPC** for communicating with the **Spectre** simulator were reused in other projects[Poi22, p. 103], and will enable further exploration of the design space when improved models become available.

6.3.1 Modeling issues

Model instabilities were a recurring issue over the course of this project. Problems encountered include convergence issues due to non-linearities, as well as divergence issues on performance metrics that do not reflect physical behavior and tend to confuse **DSE** methods. Pathological performance degradation was also encountered in edge-cases, partly due to the amount of state-tracking performed by the Landau model, that tracks turning points encountered during voltage sweeps. Automated exploration tends to systematically converge towards problematic configurations, due to their interaction with metrics extraction methods. This allowed

multiple issues to be identified and corrected within the models, which could be a valuable approach during model development, perhaps as a complementary tool to fuzzing methods.

Unfortunately, model instability prevented the use of fully automated DSE pipelines. Less accurate, but more stable ones may be employed in the future, although the impact of accuracy loss on DSE results would need to be quantified.

6.4 System-level performance evaluation

System-level performance evaluation was one of the objectives of this work, though it fell short of the goal of predicting system-level performance values from circuit-level measurements. This can be attributed to two factors: the lack of such measurements, and the vast scope of the problem.

Circuit-level data was expected to originate from DSE activities, but the low confidence in both the accuracy and optimality of these results instead often led to using values from literature, and some experimental data, though characterization of the samples is not yet complete.

The scope of system-level benchmarking encompasses the definition of new system architectures, selection of relevant benchmarks, and porting benchmarks to the selected architectures. Results are then compared to a baseline implementation on classical architectures.

The approach chosen for the benchmarking platform described in section 5.4 only considered the equivalent of a Von Neumann architecture’s unified memory, with possible in-memory accelerators. This prevented the generation of comparison points for traditional architectures, as CPU latency and energy consumption could not be tracked. While it is possible to extract such performance metrics from current-generation commercial systems, or from literature, the comparison would not take place at the same level of abstraction and optimization, reducing the confidence of comparison results.

Other benchmarking tools and simulators such as Gem5-X[Qur+21] could also help overcome this limitation, either by complementing, or by replacing the current implementation.

Nevertheless, the benchmarking platform can be leveraged for comparative evaluation of multiple accelerator architectures, and memory technologies. As a data point, an interpolator architecture was studied to determine whether it could leverage ferroelectric memory with then-current performance values, and found that this architecture would need a 80 % reduction in energy consumption to efficiently use the memory array as a LiM accelerator. Such performance improvements have since been experimentally demonstrated[Fra+21], which may enable this implementation, and others, to be competitive on some metrics.

6.5 Short-term perspectives

Perspectives abound; from the development of improved models that would facilitate automated DSE, to more characterization work on fabricated samples.

6.5.1 Remaining characterization work

The fabricated PsFeFET bitcell is interesting for designs where FeFET are not available, and where a 2T1C cell does not provide the required performance. Further characterization of this cell is expected to be performed, which will notably compare the two different FeCap orientations, bringing more insight into the metallic vias’ impact on performance.

The 2T1C cell has been studied more thoroughly, and is expected to undergo further characterization work, including its performance as a 1T1C cell, and a more detailed investigation of the 0 V readout current shown in Figure 3.7.

The image filter described in section 4.6 will also be revisited, as well as the non-pipelined multi-stage variant, to assess their potential for neuromorphic applications, such as CNNs.

6.5.2 Future simulations

DSE results presented in section 5.3.1 need to be further investigated, though they probably stem from wrongly propagated parameters, or issues with the FeCap model. A comparative

analysis of different **FeCap** compact models with their impact on accuracy for **DSE** would allow determining acceptable performance-accuracy tradeoffs for this use-case.

System level benchmarks can be revisited with more current performance values, and an in-progress implementation of a convolutional filter would, upon completion, allow exploring architectural variations, and comparing with experimental image filter results.

6.6 Future considerations for ferroelectric technology

6.6.1 Space efficiency

Larger **FeCap** designs can be explored, and footprint of current designs can be reduced by the use of optimized capacitor geometries, such as deep-trench capacitors, making the oxide layer and associated electrodes vertical instead of horizontal, therefore reducing the footprint.

Alternative space-efficient geometries are being explored, such as nanowires coated with ferroelectric oxide[[Lee+22](#)]. This could also be combined with vertical nanowire transistors[[Poi22](#)] to form vertical nanowire **FeFETs**.

Coercive voltage being highly dependent on the field orientation, controlling its direction using multiple electrodes could open alternatives to fine voltage-control for **MLC** operation. Moreover, given that crystalline domains aligned with the electric field have a lower apparent coercive voltage, this may be a practical way to store multiple bits of data per capacitor structure. Polarizing ferroelectric **HfZrO₂** however requires electric fields of a relatively high intensity of $\sim 1 \text{ MV m}^{-1}$ [[Mul+21](#); [KCJ21](#)], usually obtained by applying a low voltage across a small (about 10 nm) oxide layer. These small geometry are easier to fabricate vertically as oxide thickness is better controlled in that direction with deposition techniques than laterally with lithography and patterning techniques. The effect may nevertheless be measurable for small deviation angles.

6.6.2 Control signals

Larger ferroelectric devices have bigger E_C distributions. Assuming a device does not encounter catastrophic failures such as electrical shorts, and depending on the precise fatigue mechanisms, memory window may be maintained at the same level by progressively increasing operating voltages over the lifetime of the device, progressively making use of ferroelectric domains with higher V_C values.

As described in [subsection 2.3.1](#), it is possible to use **FeCaps** as regular capacitors by maintaining the applied voltage below V_C . As the voltage nears polarization reversal by crossing the appropriate threshold, capacitance becomes non-linear, abruptly increasing. This may provide an additional non-destructive readout mechanism for 1T1C-like structures. This effect may be less measurable in larger, multi-domain **FeCaps** where repolarization is progressive, and in devices where the ferroelectric-to-paraelectric capacitance ratio is small.

The destructive-read operation mode of multiple **FeCap** designs can be constraining, though it also offers new perspectives: read operations can be combined with a write operation at no extra cost, minimizing latency and wear if both operations need to be carried out simultaneously. They may also offer additional confidentiality guarantees in sensitive contexts by providing a mechanism to disable **WBs**, making them the opposite of **Write Once, Read Many (WORM)** memories: Write Many times, but only Read Once. Lastly, prediction mechanisms could reduce cycling caused by destructive reading of 1T1C cells: a **WB** being necessary only in cases where a **FeCap** is repolarized when attempting to determine its polarization, correctly predicting the stored data and confirming that applying the corresponding voltage does not cause a repolarization would avoid the need to write the original value back. This would only be possible if data can be predicted, and would be less effective on **MLC** memory. However, reducing the number of writes would be beneficial for read speed, as well as memory endurance.

Bibliography

- [AG21] Infineon Technologies AG. *Endurance and Data Retention Characterization of Infineon Flash Memory, Application note AN217979*. Apr. 19, 2021. URL: https://www.infineon.com/dgdl/Infineon-AN217979_Endurance_and_Data_Retention_Characterization_of_Infineon_Flash_Memory-ApplicationNotes-v03_00-EN.pdf?fileId=8ac78c8c7cdc391c017d0d30d6b064f5 (visited on 04/14/2023).
- [Alc+22] R. Alcala et al. “BEOL Integrated Ferroelectric HfO₂-Based Capacitors for FeRAM: Extrapolation of Reliability Performance to Use Conditions.” In: *IEEE Journal of the Electron Devices Society* 10 (2022). Conference Name: IEEE Journal of the Electron Devices Society, pp. 907–912. ISSN: 2168-6734. DOI: [10.1109/JEDS.2022.3198138](https://doi.org/10.1109/JEDS.2022.3198138).
- [Azi+18] A. Aziz et al. “Computing with ferroelectric FETs: Devices, models, systems, and applications.” In: *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*. Mar. 2018, pp. 1289–1298. DOI: [10.23919/DATE.2018.8342213](https://doi.org/10.23919/DATE.2018.8342213).
- [Bar90] T.J. Barnes. “SKILL: a CAD system extension language.” In: *27th ACM/IEEE Design Automation Conference*. 27th ACM/IEEE Design Automation Conference. ISSN: 0738-100X. June 1990, pp. 266–271. DOI: [10.1109/DAC.1990.114865](https://doi.org/10.1109/DAC.1990.114865).
- [BD00] William C. Black and Bodhisattva Das. “Programmable logic using giant-magnetoresistance and spin-dependent tunneling devices (invited).” In: *Journal of Applied Physics* 87 (May 2000), pp. 6674–6679. DOI: [10.1063/1.372806](https://doi.org/10.1063/1.372806).
- [BD20] Julian Blank and Kalyanmoy Deb. “pymoo: Multi-Objective Optimization in Python.” In: *IEEE Access* 8 (2020). Conference Name: IEEE Access, pp. 89497–89509. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2020.2990567](https://doi.org/10.1109/ACCESS.2020.2990567). URL: <https://ieeexplore.ieee.org/document/9078759/?arnumber=9078759> (visited on 02/02/2025).
- [Bec+18] Noah Beck et al. “‘Zeppelin’: An SoC for multichip architectures.” In: *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*. 2018 IEEE International Solid - State Circuits Conference - (ISSCC). ISSN: 2376-8606. Feb. 2018, pp. 40–42. DOI: [10.1109/ISSCC.2018.8310173](https://doi.org/10.1109/ISSCC.2018.8310173).
- [Bey+20] Sven Beyer et al. “FeFET: A versatile CMOS compatible device with game-changing potential.” en. In: *2020 IEEE International Memory Workshop (IMW)*. Dresden, Germany: IEEE, May 2020, pp. 1–4. ISBN: 978-1-72816-306-2. DOI: [10.1109/IMW48823.2020.9108150](https://doi.org/10.1109/IMW48823.2020.9108150). URL: <https://ieeexplore.ieee.org/document/9108150/> (visited on 06/21/2021).
- [BG15] S. Brisset and F. Gillon. “4 - Approaches for multi-objective optimization in the ecodeign of electric systems.” In: *Eco-Friendly Innovation in Electricity Transmission and Distribution Networks*. Ed. by Jean-Luc Bessède. Oxford: Woodhead Publishing, Jan. 1, 2015, pp. 83–97. ISBN: 978-1-78242-010-1. DOI: [10.1016/B978-1-78242-010-1.00004-5](https://doi.org/10.1016/B978-1-78242-010-1.00004-5). URL: <https://www.sciencedirect.com/science/article/pii/B9781782420101000045> (visited on 10/09/2022).
- [BI13] Mahdi Nazm Bojnordi and Engin Ipek. “DESC: energy-efficient data exchange using synchronized counters.” In: *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO-46. New York, NY, USA: Association for Computing Machinery, Dec. 7, 2013, pp. 234–246. ISBN: 978-1-4503-2638-4. DOI: [10.1145/2540708.2540729](https://doi.org/10.1145/2540708.2540729). URL: <https://doi.org/10.1145/2540708.2540729> (visited on 01/09/2023).

- [Bin+11] Nathan Binkert et al. “The gem5 simulator.” In: *ACM SIGARCH Computer Architecture News* 39.2 (Aug. 31, 2011), pp. 1–7. ISSN: 0163-5964. DOI: [10.1145/2024716.2024718](https://doi.org/10.1145/2024716.2024718). URL: <https://doi.org/10.1145/2024716.2024718> (visited on 02/11/2023).
- [Boh07] Mark Bohr. “A 30 Year Retrospective on Dennard’s MOSFET Scaling Paper.” In: *IEEE Solid-State Circuits Newsletter* 12.1 (2007), pp. 11–13. ISSN: 1098-4232. DOI: [10.1109/N-SSC.2007.4785534](https://doi.org/10.1109/N-SSC.2007.4785534). URL: <http://ieeexplore.ieee.org/document/4785534/> (visited on 12/12/2022).
- [Bös+11a] T. S. Böske et al. “Phase transitions in ferroelectric silicon doped hafnium oxide.” In: *Applied Physics Letters* 99.11 (Sept. 12, 2011). Publisher: American Institute of Physics, p. 112904. ISSN: 0003-6951. DOI: [10.1063/1.3636434](https://doi.org/10.1063/1.3636434). URL: <https://aip.scitation.org/doi/10.1063/1.3636434> (visited on 11/17/2020).
- [Bös+11b] Tim Böske et al. “Ferroelectricity in Hafnium Oxide Thin Films.” In: *Applied Physics Letters* 99 (Sept. 5, 2011), pp. 102903–102903. DOI: [10.1063/1.3634052](https://doi.org/10.1063/1.3634052).
- [Bou+19] Jordan Bouaziz et al. “Dramatic impact of pressure and annealing temperature on the properties of sputtered ferroelectric HZO layers.” In: *APL Materials* 7.8 (Aug. 2019). Publisher: American Institute of Physics, p. 081109. DOI: [10.1063/1.5110894](https://doi.org/10.1063/1.5110894). URL: <https://aip.scitation.org/doi/10.1063/1.5110894> (visited on 01/17/2023).
- [Bou20] Jordan Bouaziz. “Mémoires ferroélectriques non-volatiles à base de (Hf,Zr)O₂ pour la nanoélectronique basse consommation.” These de doctorat. Lyon, July 15, 2020. URL: <https://www.theses.fr/2020LYSEI057> (visited on 09/25/2022).
- [Bre+18] E. T. Breyer et al. “Demonstration of versatile nonvolatile logic gates in 28nm HKMG FeFET technology.” In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2018 IEEE International Symposium on Circuits and Systems (ISCAS). ISSN: 2379-447X. May 2018, pp. 1–5. DOI: [10.1109/ISCAS.2018.8351408](https://doi.org/10.1109/ISCAS.2018.8351408).
- [Bri21] Adil Brik. “Méthode de conception des systèmes intégrés multi-physique et continues-discrets.” These de doctorat. Lyon, Sept. 21, 2021. URL: <https://www.theses.fr/2021LYSEC036> (visited on 09/25/2022).
- [Cha+08] Mau-Chung Frank Chang et al. “RF interconnects for communications on-chip.” In: *Proceedings of the 2008 international symposium on Physical design*. ISPD ’08. New York, NY, USA: Association for Computing Machinery, Apr. 13, 2008, pp. 78–83. ISBN: 978-1-60558-048-7. DOI: [10.1145/1353629.1353649](https://doi.org/10.1145/1353629.1353649). URL: <https://doi.org/10.1145/1353629.1353649> (visited on 01/09/2023).
- [CHE11] Trevor E. Carlson, Wim Heirman, and Lieven Eeckhout. “Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation.” In: *SC ’11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. SC ’11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. ISSN: 2167-4337. Nov. 2011, pp. 1–12. DOI: [10.1145/2063384.2063454](https://doi.org/10.1145/2063384.2063454).
- [CL07] Premi Chandra and Peter B. Littlewood. “A Landau Primer for Ferroelectrics.” In: *Physics of Ferroelectrics: A Modern Perspective*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 69–116. ISBN: 978-3-540-34591-6. DOI: [10.1007/978-3-540-34591-6_3](https://doi.org/10.1007/978-3-540-34591-6_3). URL: https://doi.org/10.1007/978-3-540-34591-6_3.
- [CUDA17] *Nvidia CUDA Home Page*. NVIDIA Developer. July 18, 2017. URL: <https://developer.nvidia.com/cuda-zone> (visited on 02/12/2023).
- [Dav23] Schiavone Davide. *X-HEEP Github repository*. original-date: 2022-01-07T18:19:11Z. Feb. 11, 2023. URL: <https://github.com/esl-epfl/x-heep> (visited on 02/11/2023).

- [Deb01] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Google-Books-ID: OSTn4GSy2uQC. John Wiley & Sons, July 5, 2001. 540 pp. ISBN: 978-0-471-87339-6.
- [Den+20] Shan Deng et al. “A Comprehensive Model for Ferroelectric FET Capturing the Key Behaviors: Scalability, Variation, Stochasticity, and Accumulation.” In: *2020 IEEE Symposium on VLSI Technology*. 2020 IEEE Symposium on VLSI Technology. ISSN: 2158-9682. June 2020, pp. 1–2. DOI: [10.1109/VLSITechnology18217.2020.9265014](https://doi.org/10.1109/VLSITechnology18217.2020.9265014).
- [Den+22] Benoît W. Denkinger et al. “VWR2A: a very-wide-register reconfigurable-array architecture for low-power embedded devices.” In: *Proceedings of the 59th ACM/IEEE Design Automation Conference*. DAC '22. New York, NY, USA: Association for Computing Machinery, Aug. 23, 2022, pp. 895–900. ISBN: 978-1-4503-9142-9. DOI: [10.1145/3489517.3530980](https://doi.org/10.1145/3489517.3530980). URL: <https://doi.org/10.1145/3489517.3530980> (visited on 02/11/2023).
- [Den+74] R.H. Dennard et al. “Design of ion-implanted MOSFET’s with very small physical dimensions.” In: *IEEE Journal of Solid-State Circuits* 9.5 (Oct. 1974). Conference Name: IEEE Journal of Solid-State Circuits, pp. 256–268. ISSN: 1558-173X. DOI: [10.1109/JSSC.1974.1050511](https://doi.org/10.1109/JSSC.1974.1050511).
- [Dev49] A.F. Devonshire. “XCVI. Theory of barium titanate.” In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 40.309 (1949), pp. 1040–1063. DOI: [10.1080/14786444908561372](https://doi.org/10.1080/14786444908561372). eprint: <https://doi.org/10.1080/14786444908561372>. URL: <https://doi.org/10.1080/14786444908561372>.
- [Don+12] Xiangyu Dong et al. “NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory.” In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 31.7 (July 2012). Conference Name: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp. 994–1007. ISSN: 1937-4151. DOI: [10.1109/TCAD.2012.2185930](https://doi.org/10.1109/TCAD.2012.2185930).
- [Dup22] Etienne Dupuis. “Weight-sharing methods for retraining-free CNN compression.” These de doctorat. Université de Lyon, May 19, 2022. URL: <https://www.theses.fr/2022LYSEC017> (visited on 09/25/2022).
- [Dut+22] Sourav Dutta et al. “Logic Compatible High-Performance Ferroelectric Transistor Memory.” In: *IEEE Electron Device Letters* 43.3 (Mar. 2022). Conference Name: IEEE Electron Device Letters, pp. 382–385. ISSN: 1558-0563. DOI: [10.1109/LED.2022.3148669](https://doi.org/10.1109/LED.2022.3148669).
- [Esp22] Espressif. *ESP8266EX Datasheet, v6.8*. Oct. 2022. URL: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf (visited on 12/11/2022).
- [FKK12] V. Fridkin, M. Kuehn, and H. Kliem. “The Weiss model and the Landau–Khalatnikov model for the switching of ferroelectrics.” In: *Physica B: Condensed Matter* 407.12 (June 15, 2012), pp. 2211–2214. ISSN: 0921-4526. DOI: [10.1016/j.physb.2012.02.043](https://doi.org/10.1016/j.physb.2012.02.043). URL: <https://www.sciencedirect.com/science/article/pii/S0921452612002311> (visited on 02/25/2023).
- [Fra+19a] T. Francois et al. “Demonstration of BEOL-compatible ferroelectric Hf_{0.5}Zr_{0.5}O₂ scaled FeRAM co-integrated with 130nm CMOS for embedded NVM applications.” In: *2019 IEEE International Electron Devices Meeting (IEDM)*. 2019 IEEE International Electron Devices Meeting (IEDM). ISSN: 2156-017X. Dec. 2019, pp. 15.7.1–15.7.4. DOI: [10.1109/IEDM19573.2019.8993485](https://doi.org/10.1109/IEDM19573.2019.8993485).
- [Fra+19b] T. Francois et al. “Ferroelectric HfO₂ for Memory Applications: Impact of Si Doping Technique and Bias Pulse Engineering on Switching Performance.” In: *2019 IEEE 11th International Memory Workshop (IMW)*. 2019 IEEE 11th International Memory Workshop (IMW). ISSN: 2573-7503. May 2019, pp. 1–4. DOI: [10.1109/IMW.2019.8739664](https://doi.org/10.1109/IMW.2019.8739664).

- [Fra+21] T. Francois et al. “16kbit HfO₂:Si-based 1T-1C FeRAM Arrays Demonstrating High Performance Operation and Solder Reflow Compatibility.” en. In: *2021 IEEE International Electron Devices Meeting (IEDM)*. San Francisco, CA, USA: IEEE, Dec. 2021, pp. 33.1.1–33.1.4. ISBN: 978-1-66542-572-8. DOI: [10.1109/IEDM19574.2021.9720640](https://doi.org/10.1109/IEDM19574.2021.9720640). URL: <https://ieeexplore.ieee.org/document/9720640/> (visited on 07/28/2022).
- [Fra12] Felipe Frantz Ferreira. “Architectural exploration methods and tools for heterogeneous 3D-IC.” These de doctorat. Ecully, Ecole centrale de Lyon, Oct. 26, 2012. URL: <https://www.theses.fr/2012ECDL0033> (visited on 10/09/2022).
- [FS19] Shosuke Fujii and Masumi Saitoh. “Chapter 10.3 - Ferroelectric Tunnel Junction.” In: *Ferroelectricity in Doped Hafnium Oxide: Materials, Properties and Devices*. Ed. by Uwe Schroeder, Cheol Seong Hwang, and Hiroshi Funakubo. Woodhead Publishing Series in Electronic and Optical Materials. Woodhead Publishing, 2019, pp. 437–449. ISBN: 978-0-08-102430-0. DOI: <https://doi.org/10.1016/B978-0-08-102430-0.00021-8>. URL: <https://www.sciencedirect.com/science/article/pii/B9780081024300000218>.
- [Gal+19a] W.J. Gallagher et al. “22nm STT-MRAM for Reflow and Automotive Uses with High Yield, Reliability, and Magnetic Immunity and with Performance and Shielding Options.” In: *2019 IEEE International Electron Devices Meeting (IEDM)*. 2019 IEEE International Electron Devices Meeting (IEDM). ISSN: 2156-017X. Dec. 2019, pp. 2.7.1–2.7.4. DOI: [10.1109/IEDM19573.2019.8993469](https://doi.org/10.1109/IEDM19573.2019.8993469).
- [Gal+19b] W.J. Gallagher et al. “Recent Progress and Next Directions for Embedded MRAM Technology.” In: *2019 Symposium on VLSI Technology*. 2019 Symposium on VLSI Technology. ISSN: 2158-9682. June 2019, T190–T191. DOI: [10.23919/VLSIT.2019.8776547](https://doi.org/10.23919/VLSIT.2019.8776547).
- [Gas+19] C. Gastaldi et al. “Transient Negative Capacitance of Silicon-doped HfO₂ in MFIS and MFIS structures: experimental insights for hysteresis-free steep slope NC FETs.” In: *2019 IEEE International Electron Devices Meeting (IEDM)*. 2019 IEEE International Electron Devices Meeting (IEDM). ISSN: 2156-017X. Dec. 2019, pp. 23.5.1–23.5.4. DOI: [10.1109/IEDM19573.2019.8993523](https://doi.org/10.1109/IEDM19573.2019.8993523).
- [GB14] Vincent Garcia and Manuel Bibes. “Ferroelectric tunnel junctions for information storage and processing.” In: *Nature Communications* 5.1 (July 24, 2014). Number: 1 Publisher: Nature Publishing Group, p. 4289. ISSN: 2041-1723. DOI: [10.1038/ncomms5289](https://doi.org/10.1038/ncomms5289). URL: <https://www.nature.com/articles/ncomms5289/> (visited on 02/25/2023).
- [Gir+21] Patrick Girard et al. “A Survey of Test and Reliability Solutions for Magnetic Random Access Memories.” In: *Proceedings of the IEEE* 109.2 (Feb. 2021). Conference Name: Proceedings of the IEEE, pp. 149–169. ISSN: 1558-2256. DOI: [10.1109/JPROC.2020.3029600](https://doi.org/10.1109/JPROC.2020.3029600).
- [Giu21] Gino Giusi. “Floating Body DRAM with Body Raised and Source/Drain Separation.” In: *Electronics* 10.6 (Jan. 2021). Number: 6 Publisher: Multidisciplinary Digital Publishing Institute, p. 706. ISSN: 2079-9292. DOI: [10.3390/electronics10060706](https://doi.org/10.3390/electronics10060706). URL: <https://www.mdpi.com/2079-9292/10/6/706> (visited on 09/15/2022).
- [Gre+20] L. Grenouillet et al. “Performance assessment of BEOL-integrated HfO₂-based ferroelectric capacitors for FeRAM memory arrays.” In: *2020 IEEE Silicon Nanoelectronics Workshop (SNW)*. 2020 IEEE Silicon Nanoelectronics Workshop (SNW). ISSN: 2161-4644. June 2020, pp. 5–6. DOI: [10.1109/SNW50361.2020.9131648](https://doi.org/10.1109/SNW50361.2020.9131648).
- [HCMOS] *EUROPRACTICE | STMicroelectronics*. URL: <https://europactice-ic.com/technologies/asics/stmicroelectronics/> (visited on 02/24/2023).
- [HIP23] *HIP: C++ Heterogeneous-Compute Interface for Portability*. original-date: 2016-01-07T17:41:56Z. Feb. 10, 2023. URL: <https://github.com/ROCm-Developer-Tools/HIP> (visited on 02/12/2023).

- [HKMG20] *28nm HKMG Technologies / GLOBALFOUNDRIES*. Dec. 9, 2020. URL: <https://web.archive.org/web/20201209211036/https://www.globalfoundries.com/technology-solutions/cmos/fdx/28nm-hkmg-technologies> (visited on 02/24/2023).
- [Hon+01] Seungbum Hong et al. “Principle of ferroelectric domain imaging using atomic force microscope.” en. In: *Journal of Applied Physics* 89.2 (Jan. 2001), pp. 1377–1386. ISSN: 0021-8979, 1089-7550. DOI: [10.1063/1.1331654](https://doi.org/10.1063/1.1331654). URL: <http://aip.scitation.org/doi/10.1063/1.1331654> (visited on 07/28/2022).
- [Ihl19] Jon F. Ihlefeld. “Chapter 1 - Fundamentals of Ferroelectric and Piezoelectric Properties.” In: *Ferroelectricity in Doped Hafnium Oxide: Materials, Properties and Devices*. Ed. by Uwe Schroeder, Cheol Seong Hwang, and Hiroshi Funakubo. Woodhead Publishing Series in Electronic and Optical Materials. Woodhead Publishing, 2019, pp. 1–24. ISBN: 978-0-08-102430-0. DOI: <https://doi.org/10.1016/B978-0-08-102430-0.00001-2>. URL: <https://www.sciencedirect.com/science/article/pii/B9780081024300000012>.
- [Ike+20] Sumio Ikegawa et al. “Magnetoresistive Random Access Memory: Present and Future.” In: *IEEE Transactions on Electron Devices* 67.4 (Apr. 2020). Conference Name: IEEE Transactions on Electron Devices, pp. 1407–1419. ISSN: 1557-9646. DOI: [10.1109/TED.2020.2965403](https://doi.org/10.1109/TED.2020.2965403).
- [IRDS22] *International Roadmap for Devices and Systems (IRDS™) 2022 Edition - IEEE IRDS™*. 2022. URL: <https://irds.ieee.org/editions/2022> (visited on 02/25/2023).
- [J14] Matt J. *Simple 1D polynomial fitting with particular coefficients constrained to zero*. Mar. 2014. URL: https://www.mathworks.com/matlabcentral/answers/123072-curve-fitting-tool-with-custom-equation-odd-power-polynomial#answer_130371 (visited on 06/21/2021).
- [Jao+21] Nicholas Jao et al. “Design Space Exploration of Ferroelectric Tunnel Junction Toward Crossbar Memories.” In: *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits* 7.2 (Dec. 1, 2021). Publisher: Institute of Electrical and Electronics Engineers. ISSN: 2329-9231. DOI: [10.1109/JXCDC.2021.3117566](https://doi.org/10.1109/JXCDC.2021.3117566). URL: <https://www.osti.gov/pages/biblio/1829766> (visited on 02/22/2023).
- [Jer+17] Matthew Jerry et al. “Ferroelectric FET analog synapse for acceleration of deep neural network training.” In: *2017 IEEE International Electron Devices Meeting (IEDM)*. 2017 IEEE International Electron Devices Meeting (IEDM). ISSN: 2156-017X. Dec. 2017, pp. 6.2.1–6.2.4. DOI: [10.1109/IEDM.2017.8268338](https://doi.org/10.1109/IEDM.2017.8268338).
- [JGL16] Mansi Jhamb, Garima, and Himanshu Lohani. “Design, implementation and performance comparison of multiplier topologies in power-delay space.” en. In: *Engineering Science and Technology, an International Journal* 19.1 (Mar. 2016), pp. 355–363. ISSN: 22150986. DOI: [10.1016/j.jestch.2015.08.006](https://doi.org/10.1016/j.jestch.2015.08.006). URL: <https://linkinghub.elsevier.com/retrieve/pii/S2215098615001287> (visited on 01/31/2020).
- [Kap13] Kapooht. *Von Neumann Architecture Diagramm*. Apr. 28, 2013. URL: https://commons.wikimedia.org/wiki/File:Von_Neumann_Architecture.svg (visited on 12/22/2022).
- [Kaz+21a] Arman Kazemi et al. “In-Memory Nearest Neighbor Search with FeFET Multi-Bit Content-Addressable Memories.” In: *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE). ISSN: 1558-1101. Feb. 2021, pp. 1084–1089. DOI: [10.23919/DAT51398.2021.9474025](https://doi.org/10.23919/DAT51398.2021.9474025).
- [Kaz+21b] Arman Kazemi et al. “MIMHD: Accurate and Efficient Hyperdimensional Inference Using Multi-Bit In-Memory Computing.” In: *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. 2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). July 2021, pp. 1–6. DOI: [10.1109/ISLPED52811.2021.9502498](https://doi.org/10.1109/ISLPED52811.2021.9502498).

- [Kaz+22] Arman Kazemi et al. “Achieving software-equivalent accuracy for hyperdimensional computing with ferroelectric-based in-memory computing.” In: *Scientific Reports* 12.1 (Nov. 10, 2022). Number: 1 Publisher: Nature Publishing Group, p. 19201. ISSN: 2045-2322. DOI: [10.1038/s41598-022-23116-w](https://doi.org/10.1038/s41598-022-23116-w). URL: <https://www.nature.com/articles/s41598-022-23116-w> (visited on 02/11/2023).
- [KCJ21] Jae Young Kim, Min-Ju Choi, and Ho Won Jang. “Ferroelectric field effect transistors: Progress and perspective.” In: *APL Materials* 9.2 (Feb. 2021). Publisher: American Institute of Physics, p. 021102. DOI: [10.1063/5.0035515](https://doi.org/10.1063/5.0035515). URL: <https://aip.scitation.org/doi/10.1063/5.0035515> (visited on 04/16/2023).
- [Kle+21] Dominik Kleimaier et al. “Demonstration of a p-Type Ferroelectric FET With Immediate Read-After-Write Capability.” In: *IEEE Electron Device Letters* 42.12 (Dec. 2021). Conference Name: IEEE Electron Device Letters, pp. 1774–1777. ISSN: 1558-0563. DOI: [10.1109/LED.2021.3118645](https://doi.org/10.1109/LED.2021.3118645).
- [KN03] Hiroaki Kato and Hiroshi Nozawa. “Proposal for 1T/1C Ferroelectric Random Access Memory with Multiple Storage and Application to Functional Memory.” In: *Japanese Journal of Applied Physics* 42 (Sept. 2003), pp. 5998–6002.
- [Knu70] Donald E. Knuth. “Von Neumann’s First Computer Program.” In: *ACM Computing Surveys* 2.4 (Dec. 1, 1970), pp. 247–260. ISSN: 0360-0300. DOI: [10.1145/356580.356581](https://doi.org/10.1145/356580.356581). URL: <https://doi.org/10.1145/356580.356581> (visited on 01/10/2023).
- [Koo+18] Maha Kooli et al. “Smart instruction codes for in-memory computing architectures compatible with standard SRAM interfaces.” In: *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE). ISSN: 1558-1101. Mar. 2018, pp. 1634–1639. DOI: [10.23919/DATE.2018.8342276](https://doi.org/10.23919/DATE.2018.8342276).
- [Led+20] Maximilian Lederer et al. “Structural and Electrical Comparison of Si and Zr Doped Hafnium Oxide Thin Films and Integrated FeFETs Utilizing Transmission Kikuchi Diffraction.” In: *Nanomaterials* 10.2 (Feb. 2020). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 384. ISSN: 2079-4991. DOI: [10.3390/nano10020384](https://doi.org/10.3390/nano10020384). URL: <https://www.mdpi.com/2079-4991/10/2/384> (visited on 02/22/2023).
- [Led+21] M. Lederer et al. “Impact of the SiO₂ interface layer on the crystallographic texture of ferroelectric hafnium oxide.” In: *Applied Physics Letters* 118.1 (Jan. 4, 2021), p. 012901. ISSN: 0003-6951, 1077-3118. DOI: [10.1063/5.0029635](https://doi.org/10.1063/5.0029635). URL: <https://aip.scitation.org/doi/10.1063/5.0029635> (visited on 02/22/2023).
- [Lee+22] Dong Hyun Lee et al. “Neuromorphic devices based on fluorite-structured ferroelectrics.” In: *InfoMat* 4.12 (Dec. 2022). ISSN: 2567-3165, 2567-3165. DOI: [10.1002/inf2.12380](https://doi.org/10.1002/inf2.12380). URL: <https://onlinelibrary.wiley.com/doi/10.1002/inf2.12380> (visited on 04/24/2023).
- [Leh+21] David Lehninger 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). Kyoto, Japan: IEEE, July 6, 2021, pp. 1–4. ISBN: 978-1-72817-632-1. DOI: [10.1109/IITC51362.2021.9537346](https://doi.org/10.1109/IITC51362.2021.9537346). URL: <https://ieeexplore.ieee.org/document/9537346/> (visited on 07/05/2022).
- [LFZ11] Zhichao Lu, Jerry G. Fossum, and Zhenming Zhou. “A Floating-Body/Gate DRAM Cell Upgraded for Long Retention Time.” In: *IEEE Electron Device Letters* 32.6 (June 2011). Conference Name: IEEE Electron Device Letters, pp. 731–733. ISSN: 1558-0563. DOI: [10.1109/LED.2011.2134065](https://doi.org/10.1109/LED.2011.2134065).
- [LHS22] You-Sheng Liu, Yuan-Yu Huang, and Pin Su. “Design Space Exploration for Scaled FeFET Nonvolatile Memories: High-k Spacer as a Powerful Aid.” In: *2022 6th IEEE Electron Devices Technology & Manufacturing Conference (EDTM)*. 2022 6th IEEE Electron Devices Technology & Manufacturing Conference (EDTM). Mar. 2022, pp. 70–72. DOI: [10.1109/EDTM53872.2022.9798076](https://doi.org/10.1109/EDTM53872.2022.9798076).

- [Li+09] Sheng Li et al. “McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures.” In: *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO 42. New York, NY, USA: Association for Computing Machinery, Dec. 12, 2009, pp. 469–480. ISBN: 978-1-60558-798-1. DOI: [10.1145/1669112.1669172](https://doi.org/10.1145/1669112.1669172). URL: <https://doi.org/10.1145/1669112.1669172> (visited on 02/11/2023).
- [Liu+14] Wulong Liu et al. “Exploration of Electrical and Novel Optical Chip-to-Chip Interconnects.” In: *IEEE Design & Test* 31.5 (Oct. 2014), pp. 28–35. ISSN: 2168-2356, 2168-2364. DOI: [10.1109/MDAT.2014.2336217](https://doi.org/10.1109/MDAT.2014.2336217). URL: <http://ieeexplore.ieee.org/document/6849444/> (visited on 02/25/2023).
- [Mag+02] P.S. Magnusson et al. “Simics: A full system simulation platform.” In: *Computer* 35.2 (Feb. 2002). Conference Name: Computer, pp. 50–58. ISSN: 1558-0814. DOI: [10.1109/2.982916](https://doi.org/10.1109/2.982916).
- [Maj+18] Sayani Majumdar et al. “Electrode Dependence of Tunneling Electroresistance and Switching Stability in Organic Ferroelectric P(VDF-TrFE)-Based Tunnel Junctions.” In: *Advanced Functional Materials* 28.15 (2018). _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/adfm.201703273>, p. 1703273. ISSN: 1616-3028. DOI: [10.1002/adfm.201703273](https://doi.org/10.1002/adfm.201703273). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adfm.201703273> (visited on 02/25/2023).
- [Maj22] Sayani Majumdar. “Back-End CMOS Compatible and Flexible Ferroelectric Memories for Neuromorphic Computing and Adaptive Sensing.” In: *Advanced Intelligent Systems* 4.4 (Apr. 2022), p. 2100175. ISSN: 2640-4567, 2640-4567. DOI: [10.1002/aisy.202100175](https://doi.org/10.1002/aisy.202100175). URL: <https://onlinelibrary.wiley.com/doi/10.1002/aisy.202100175> (visited on 07/05/2022).
- [Mam+21] K vin Mambu et al. “Instruction Set Design Methodology for In-Memory Computing through QEMU-based System Emulator.” In: *2021 IEEE International Workshop on Rapid System Prototyping (RSP)*. 2021 IEEE International Workshop on Rapid System Prototyping (RSP). ISSN: 2150-5519. Oct. 2021, pp. 43–49. DOI: [10.1109/RSP53691.2021.9806255](https://doi.org/10.1109/RSP53691.2021.9806255).
- [Mar+21] C dric Marchand et al. “FeFET based Logic-in-Memory: an overview.” In: *2021 16th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*. 2021 16th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS). June 2021, pp. 1–6. DOI: [10.1109/DTIS53253.2021.9505078](https://doi.org/10.1109/DTIS53253.2021.9505078).
- [Mar+22] C dric 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 (June 1, 2022), pp. 1–1. DOI: [10.1109/JXDC.2022.3168057](https://doi.org/10.1109/JXDC.2022.3168057).
- [Mas+21] A. G. Maslovskaya et al. “Theoretical and numerical analysis of the Landau–Khalatnikov model of ferroelectric hysteresis.” In: *Communications in Nonlinear Science and Numerical Simulation* 93 (Feb. 1, 2021), p. 105524. ISSN: 1007-5704. DOI: [10.1016/j.cnsns.2020.105524](https://doi.org/10.1016/j.cnsns.2020.105524). URL: <https://www.sciencedirect.com/science/article/pii/S1007570420303543> (visited on 02/25/2023).
- [MG19] T.P. Ma and Nanbo Gong. “Retention and Endurance of FeFET Memory Cells.” In: *2019 IEEE 11th International Memory Workshop (IMW)*. 2019 IEEE 11th International Memory Workshop (IMW). ISSN: 2573-7503. May 2019, pp. 1–4. DOI: [10.1109/IMW.2019.8739726](https://doi.org/10.1109/IMW.2019.8739726).
- [Mik+19] T. Mikolajick et al. “Next Generation Ferroelectric Memories enabled by Hafnium Oxide.” In: *2019 IEEE International Electron Devices Meeting (IEDM)*. 2019 IEEE International Electron Devices Meeting (IEDM). ISSN: 2156-017X. Dec. 2019, pp. 15.5.1–15.5.4. DOI: [10.1109/IEDM19573.2019.8993447](https://doi.org/10.1109/IEDM19573.2019.8993447).
- [Mil+90] S. L. Miller et al. “Device modeling of ferroelectric capacitors.” In: *Journal of Applied Physics* 68.12 (Dec. 15, 1990). Publisher: American Institute of Physics, pp. 6463–6471. ISSN: 0021-8979. DOI: [10.1063/1.346845](https://doi.org/10.1063/1.346845). URL: <https://aip.scitation.org/doi/10.1063/1.346845> (visited on 02/26/2023).

- [Moo65] Gordon E. Moore. “Cramming more components onto integrated circuits.” In: *Electronics* 38.8 (1965), pp. 114–117. URL: <https://newsroom.intel.com/wp-content/uploads/sites/11/2018/05/moores-law-electronics.pdf>.
- [Moz21] Luca Mozzone. “Design of a benchmarking platform for Logic-In-Memory architectures based on ferroelectric HfO₂.” laurea. Politecnico di Torino, Apr. 16, 2021. 69 pp. URL: <https://webthesis.biblio.polito.it/17853/> (visited on 09/25/2022).
- [Mue+13a] Stefan Mueller et al. “From MFM Capacitors Toward Ferroelectric Transistors: Endurance and Disturb Characteristics of HfO₂-Based FeFET Devices.” In: *IEEE Transactions on Electron Devices* 60.12 (Dec. 2013). Conference Name: IEEE Transactions on Electron Devices, pp. 4199–4205. ISSN: 1557-9646. DOI: [10.1109/TED.2013.2283465](https://doi.org/10.1109/TED.2013.2283465).
- [Mue+13b] Stefan Mueller et al. “Reliability Characteristics of Ferroelectric Si : HfO₂ Thin Films for Memory Applications.” In: *IEEE Transactions on Device and Materials Reliability* 13.1 (Mar. 2013). Conference Name: IEEE Transactions on Device and Materials Reliability, pp. 93–97. ISSN: 1558-2574. DOI: [10.1109/TDMR.2012.2216269](https://doi.org/10.1109/TDMR.2012.2216269).
- [Mül+11] Johannes Müller et al. “Ferroelectric Zr_{0.5}Hf_{0.5}O₂ thin films for nonvolatile memory applications.” In: *Applied Physics Letters* 99 (Sept. 1, 2011), pp. 112901–112901. DOI: [10.1063/1.3636417](https://doi.org/10.1063/1.3636417).
- [Mül+15] J. Müller et al. “Ferroelectric Hafnium Oxide Based Materials and Devices: Assessment of Current Status and Future Prospects.” In: *ECS Journal of Solid State Science and Technology* 4.5 (2015), N30–N35. ISSN: 2162-8769, 2162-8777. DOI: [10.1149/2.0081505jss](https://doi.org/10.1149/2.0081505jss). URL: <http://jss.ecsdl.org/lookup/doi/10.1149/2.0081505jss> (visited on 04/01/2019).
- [Mul+17] H. Mulaosmanovic et al. “Novel ferroelectric FET based synapse for neuromorphic systems.” In: *2017 Symposium on VLSI Technology*. 2017 Symposium on VLSI Technology. ISSN: 2158-9682. June 2017, T176–T177. DOI: [10.23919/VLSIT.2017.7998165](https://doi.org/10.23919/VLSIT.2017.7998165).
- [Mul+21] Halid Mulaosmanovic et al. “Ferroelectric field-effect transistors based on HfO₂ : a review.” In: *Nanotechnology* 32.50 (Dec. 10, 2021), p. 502002. ISSN: 0957-4484, 1361-6528. DOI: [10.1088/1361-6528/ac189f](https://doi.org/10.1088/1361-6528/ac189f). URL: <https://iopscience.iop.org/article/10.1088/1361-6528/ac189f> (visited on 02/14/2023).
- [Mül+21] S. Müller et al. “Development status of gate-first FeFET technology.” In: *2021 Symposium on VLSI Technology*. 2021 Symposium on VLSI Technology. ISSN: 2158-9682. June 2021, pp. 1–2.
- [Ni+18] K. Ni et al. “SoC Logic Compatible Multi-Bit FeMFET Weight Cell for Neuromorphic Applications.” In: *2018 IEEE International Electron Devices Meeting (IEDM)*. 2018 IEEE International Electron Devices Meeting (IEDM). ISSN: 2156-017X. Dec. 2018, pp. 13.2.1–13.2.4. DOI: [10.1109/IEDM.2018.8614496](https://doi.org/10.1109/IEDM.2018.8614496).
- [Ni+19] Kai Ni et al. “Ferroelectric ternary content-addressable memory for one-shot learning.” In: *Nature Electronics* 2.11 (Nov. 2019). Number: 11 Publisher: Nature Publishing Group, pp. 521–529. ISSN: 2520-1131. DOI: [10.1038/s41928-019-0321-3](https://doi.org/10.1038/s41928-019-0321-3). URL: <https://www.nature.com/articles/s41928-019-0321-3> (visited on 02/12/2023).
- [Nie+23] Michael Niemier et al. “Cross Layer Design for the Predictive Assessment of Technology-Enabled Architectures.pdf.” In: *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE). Apr. 18, 2023.
- [Now+16] Janusz J. Nowak et al. “Dependence of Voltage and Size on Write Error Rates in Spin-Transfer Torque Magnetic Random-Access Memory.” In: *IEEE Magnetism Letters* 7 (2016). Conference Name: IEEE Magnetism Letters, pp. 1–4. ISSN: 1949-3088. DOI: [10.1109/LMAG.2016.2539256](https://doi.org/10.1109/LMAG.2016.2539256).

- [OCo+18] Ian O'Connor et al. "Prospects for energy-efficient edge computing with integrated HfO₂-based ferroelectric devices." In: *2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*. Oct. 2018, pp. 180–183. DOI: [10.1109/VLSI-SoC.2018.8644809](https://doi.org/10.1109/VLSI-SoC.2018.8644809).
- [Oku+21] Jun Okuno et al. "High-Endurance and Low-Voltage operation of 1T1C FeRAM Arrays for Nonvolatile Memory Application." In: *2021 IEEE International Memory Workshop (IMW)*. 2021 IEEE International Memory Workshop (IMW). ISSN: 2573-7503. May 2021, pp. 1–3. DOI: [10.1109/IMW51353.2021.9439595](https://doi.org/10.1109/IMW51353.2021.9439595).
- [OMP] *The OpenMP API specification for parallel programming Home Page*. OpenMP. URL: <https://www.openmp.org/> (visited on 02/12/2023).
- [Pal+18] Ashish Pal et al. "Scaling NC-FinFET to Sub-3 nm Nodes." In: *2018 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. 2018 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S). ISSN: 2573-5926. Oct. 2018, pp. 1–2. DOI: [10.1109/S3S.2018.8640184](https://doi.org/10.1109/S3S.2018.8640184).
- [Paw22] Richard Pawson. "The Myth of the Harvard Architecture." In: *IEEE Annals of the History of Computing* 44.3 (July 2022). Conference Name: IEEE Annals of the History of Computing, pp. 59–69. ISSN: 1934-1547. DOI: [10.1109/MAHC.2022.3175612](https://doi.org/10.1109/MAHC.2022.3175612).
- [Pen+22] Lillian Pentecost et al. "NVMEexplorer: A Framework for Cross-Stack Comparisons of Embedded Non-Volatile Memories." In: *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA). ISSN: 2378-203X. Apr. 2022, pp. 938–956. DOI: [10.1109/HPCA53966.2022.00073](https://doi.org/10.1109/HPCA53966.2022.00073).
- [Peš+17] Milan Pešić et al. "A computational study of hafnia-based ferroelectric memories: from ab initio via physical modeling to circuit models of ferroelectric device." In: *Journal of Computational Electronics* 16.4 (Dec. 1, 2017), pp. 1236–1256. ISSN: 1572-8137. DOI: [10.1007/s10825-017-1053-0](https://doi.org/10.1007/s10825-017-1053-0). URL: <https://doi.org/10.1007/s10825-017-1053-0> (visited on 09/19/2022).
- [PLH21] Hyeon Woo Park, Jae-Gil Lee, and Cheol Seong Hwang. "Review of ferroelectric field-effect transistors for three-dimensional storage applications." In: *Nano Select* 2.6 (2021). __eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/nano.202000281>, pp. 1187–1207. ISSN: 2688-4011. DOI: [10.1002/nano.202000281](https://doi.org/10.1002/nano.202000281). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nano.202000281> (visited on 02/25/2023).
- [Poi22] Arnaud Poittevin. "Logic circuits based on Vertical nanowire FETs : Physical and circuit design challenges and opportunities." These de doctorat. Lyon, June 23, 2022. URL: <https://www.theses.fr/2022LYSEC024> (visited on 02/23/2023).
- [Por+15] Matt Poremba et al. "DESTINY: A tool for modeling emerging 3D NVM and eDRAM caches." In: *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE). ISSN: 1558-1101. Mar. 2015, pp. 1543–1546. DOI: [10.7873/DATE.2015.0733](https://doi.org/10.7873/DATE.2015.0733).
- [Pre35] F. Preisach. "Über die magnetische Nachwirkung." In: *Zeitschrift für Physik* 94.5 (May 1, 1935), pp. 277–302. ISSN: 0044-3328. DOI: [10.1007/BF01349418](https://doi.org/10.1007/BF01349418). URL: <https://doi.org/10.1007/BF01349418> (visited on 09/20/2022).
- [Qur+21] Yasir Mahmood Qureshi et al. "Gem5-X: A Many-core Heterogeneous Simulation Platform for Architectural Exploration and Optimization." In: *ACM Transactions on Architecture and Code Optimization* 18.4 (July 17, 2021), 44:1–44:27. ISSN: 1544-3566. DOI: [10.1145/3461662](https://doi.org/10.1145/3461662). URL: <https://doi.org/10.1145/3461662> (visited on 02/11/2023).
- [Rag+17] Jonathan Ragan-Kelley et al. "Halide: decoupling algorithms from schedules for high-performance image processing." In: *Communications of the ACM* 61.1 (Dec. 27, 2017), pp. 106–115. ISSN: 0001-0782, 1557-7317. DOI: [10.1145/3150211](https://doi.org/10.1145/3150211). URL: <https://dl.acm.org/doi/10.1145/3150211> (visited on 02/12/2023).

- [Rav+19] Taras Ravsher et al. “Adoption of 2T2C ferroelectric memory cells for logic operation.” In: *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS). Genoa, Italy: IEEE, Nov. 2019, pp. 791–794. ISBN: 978-1-72810-996-1. DOI: [10.1109/ICECS46596.2019.8965155](https://doi.org/10.1109/ICECS46596.2019.8965155). URL: <https://ieeexplore.ieee.org/document/8965155/> (visited on 06/21/2021).
- [Rei+19] Dayane Reis et al. “Design and Analysis of an Ultra-Dense, Low-Leakage, and Fast FeFET-Based Random Access Memory Array.” In: *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits* PP (July 22, 2019), pp. 1–1. DOI: [10.1109/JXCDC.2019.2930284](https://doi.org/10.1109/JXCDC.2019.2930284).
- [Sch22] David Schor. *IEDM 2022: Did We Just Witness The Death Of SRAM?* WikiChip Fuse. Section: Foundries. Dec. 14, 2022. URL: <https://fuse.wikichip.org/news/7343/iedm-2022-did-we-just-witness-the-death-of-sram/> (visited on 01/09/2023).
- [SD08] Sayeef Salahuddin and Supriyo Datta. “Use of Negative Capacitance to Provide Voltage Amplification for Low Power Nanoscale Devices.” In: *Nano Letters* 8.2 (Feb. 2008), pp. 405–410. ISSN: 1530-6984. DOI: [10.1021/nl071804g](https://doi.org/10.1021/nl071804g). URL: <https://doi.org/10.1021/nl071804g>.
- [SG96] A. Sheikholeslami and P.G. Gulak. “Transient modeling of ferroelectric capacitors for nonvolatile memories.” In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 43.3 (May 1996). Conference Name: IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, pp. 450–456. ISSN: 1525-8955. DOI: [10.1109/58.489404](https://doi.org/10.1109/58.489404).
- [SHF19] Uwe Schroeder, Cheol Seong Hwang, and Hiroshi Funakubo. “Preface.” In: *Ferroelectricity in Doped Hafnium Oxide: Materials, Properties and Devices*. Ed. by Uwe Schroeder, Cheol Seong Hwang, and Hiroshi Funakubo. Woodhead Publishing Series in Electronic and Optical Materials. Woodhead Publishing, 2019, pp. xvii–xviii. ISBN: 978-0-08-102430-0. DOI: <https://doi.org/10.1016/B978-0-08-102430-0.09987-3>. URL: <https://www.sciencedirect.com/science/article/pii/B9780081024300099873>.
- [Sin+] Deepa Sinha et al. “New Design for Low Power High Performance 8T Full Adder.” en. In: (), p. 4.
- [Sle+19a] Stefan Slesazek et al. “A 2TnC ferroelectric memory gain cell suitable for compute-in-memory and neuromorphic application.” In: *2019 IEEE International Electron Devices Meeting (IEDM)*. 2019 IEEE International Electron Devices Meeting (IEDM). San Francisco, CA, USA: IEEE, Dec. 2019, pp. 38.6.1–38.6.4. ISBN: 978-1-72814-032-2. DOI: [10.1109/IEDM19573.2019.8993663](https://doi.org/10.1109/IEDM19573.2019.8993663). URL: <https://ieeexplore.ieee.org/document/8993663/> (visited on 06/21/2021).
- [Sle+19b] Stefan Slesazek et al. “Uniting The Trinity of Ferroelectric HfO₂ Memory Devices in a Single Memory Cell.” In: *2019 IEEE 11th International Memory Workshop (IMW)*. Monterey, CA, USA: IEEE, May 2019, pp. 1–4. ISBN: 978-1-72810-981-7. DOI: [10.1109/IMW.2019.8739742](https://doi.org/10.1109/IMW.2019.8739742). URL: <https://ieeexplore.ieee.org/document/8739742/> (visited on 01/12/2020).
- [SP21] Stefan Slesazek and Milan PESIC. “Ferroelectric memory and logic cell and operation method.” U.S. pat. 11205467B2. Namlab GmbH. Dec. 21, 2021. URL: <https://patents.google.com/patent/US11205467B2/en> (visited on 04/28/2023).
- [SR17] Ankit Sharma and Kaushik Roy. “Design Space Exploration of Hysteresis-Free HfZrOx-Based Negative Capacitance FETs.” In: *IEEE Electron Device Letters* 38.8 (Aug. 2017). Conference Name: IEEE Electron Device Letters, pp. 1165–1167. ISSN: 1558-0563. DOI: [10.1109/LED.2017.2714659](https://doi.org/10.1109/LED.2017.2714659).
- [SYCL14] *SYCL - C++ Single-source Heterogeneous Programming for Acceleration Of-fload*. The Khronos Group. Section: API. Jan. 20, 2014. URL: <https://www.khronos.org/sycl/> (visited on 02/12/2023).

- [Syn21] Synopsys. *What is Design Space Optimization (DSO)? – How It Works?* / Synopsys. Oct. 12, 2021. URL: <https://www.synopsys.com/glossary/what-is-design-space-optimization.html> (visited on 09/26/2022).
- [Vog10] Thomas Vogelsang. “Understanding the Energy Consumption of Dynamic Random Access Memories.” In: *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*. Atlanta, GA, USA: IEEE, Dec. 2010, pp. 363–374. ISBN: 978-1-4244-9071-4. DOI: [10.1109/MICRO.2010.42](https://doi.org/10.1109/MICRO.2010.42). URL: <http://ieeexplore.ieee.org/document/5695550/> (visited on 01/31/2020).
- [WA17] Muhammad Abdul Wahab and Muhammad A. Alam. *A Verilog-A Compact Model for Negative Capacitance FET*. Nov. 2017. DOI: [doi:/10.4231/D3QZ22K3Z](https://doi.org/10.4231/D3QZ22K3Z). URL: <https://nanohub.org/publications/95/5>.
- [Wat16] Andrew Waterman. “Design of the RISC-V Instruction Set Architecture.” PhD thesis. EECS Department, University of California, Berkeley, Jan. 2016. URL: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-1.html>.
- [Wil] Stephen Williams. *Icarus Verilog Home Page*. URL: <http://iverilog.icarus.com/> (visited on 02/12/2023).
- [Win+20] Jasper de Winkel et al. “Battery-Free Game Boy.” In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4.3 (Sept. 4, 2020), pp. 1–34. ISSN: 2474-9567. DOI: [10.1145/3411839](https://doi.org/10.1145/3411839). URL: <https://dl.acm.org/doi/10.1145/3411839> (visited on 03/10/2023).
- [WSW00] Robert M. Wallace, Richard A. Stoltz, and Glen D. Wilk. “Zirconium and/or hafnium oxynitride gate dielectric.” U.S. pat. 6013553A. Texas Instruments Inc. Jan. 11, 2000. URL: <https://patents.google.com/patent/US6013553A/en> (visited on 11/21/2022).
- [Wuu+22] John Wu et al. “3D V-Cache: the Implementation of a Hybrid-Bonded 64MB Stacked Cache for a 7nm x86-64 CPU.” In: *2022 IEEE International Solid-State Circuits Conference (ISSCC)*. 2022 IEEE International Solid-State Circuits Conference (ISSCC). Vol. 65. ISSN: 2376-8606. Feb. 2022, pp. 428–429. DOI: [10.1109/ISSCC42614.2022.9731565](https://doi.org/10.1109/ISSCC42614.2022.9731565).
- [Yin+16] Xunzhao Yin et al. “Exploiting ferroelectric FETs for low-power non-volatile logic-in-memory circuits.” en. In: *Proceedings of the 35th International Conference on Computer-Aided Design - ICCAD '16*. Austin, Texas: ACM Press, 2016, pp. 1–8. ISBN: 978-1-4503-4466-1. DOI: [10.1145/2966986.2967037](https://doi.org/10.1145/2966986.2967037). URL: <http://dl.acm.org/citation.cfm?doid=2966986.2967037> (visited on 04/01/2019).
- [Yin+19] Xunzhao Yin et al. “An Ultra-Dense 2FeFET TCAM Design Based on a Multi-Domain FeFET Model.” In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 66.9 (Sept. 2019). Conference Name: IEEE Transactions on Circuits and Systems II: Express Briefs, pp. 1577–1581. ISSN: 1558-3791. DOI: [10.1109/TCSII.2018.2889225](https://doi.org/10.1109/TCSII.2018.2889225).
- [Yoo+19] Insik Yoon et al. “Design space exploration of Ferroelectric FET based Processing-in-Memory DNN Accelerator.” In: *ArXiv* (Aug. 12, 2019). URL: <https://www.semanticscholar.org/paper/Design-space-exploration-of-Ferroelectric-FET-based-Yoon-Jerry/fd7d0de5ed04be930f5a4afb3c71d51a1001f967> (visited on 02/22/2023).
- [YS17] Wei-Xiang You and Pin Su. “Design Space Exploration Considering Back-Gate Biasing Effects for 2D Negative-Capacitance Field-Effect Transistors.” In: *IEEE Transactions on Electron Devices* 64.8 (Aug. 2017). Conference Name: IEEE Transactions on Electron Devices, pp. 3476–3481. ISSN: 1557-9646. DOI: [10.1109/TED.2017.2714687](https://doi.org/10.1109/TED.2017.2714687).
- [Zac+22] Christina Zacharaki et al. “Hf_{0.5}Zr_{0.5}O₂-Based Germanium Ferroelectric p-FETs for Nonvolatile Memory Applications.” In: *ACS Applied Electronic Materials* 4.6 (June 28, 2022). Publisher: American Chemical Society, pp. 2815–2821. DOI: [10.1021/acsaelm.2c00324](https://doi.org/10.1021/acsaelm.2c00324). URL: <https://doi.org/10.1021/acsaelm.2c00324> (visited on 01/17/2023).

- [Zho+20] Haidi Zhou et al. “Endurance and targeted programming behavior of HfO₂-FeFETs.” In: *2020 IEEE International Memory Workshop (IMW)*. 2020 IEEE International Memory Workshop (IMW). ISSN: 2573-7503. May 2020, pp. 1–4. DOI: [10.1109/IMW48823.2020.9108131](https://doi.org/10.1109/IMW48823.2020.9108131).

Glossary

28SLP GlobalFoundries 28 nm Super Low Power technology based on **high-k** dielectrics with metal gates[HKMG20]. 58, 83, 89, 125

3εFERRO European project funded by European Union’s Horizon 2020 research and innovation programme under grant agreement number 780302, and funding the present work. 5, 6, 15, 16, 21, 47, 49, 54, 92, 159–161

ASIC Application-Specific Integrated Circuit, Integrated Circuit that has been designed for a specific purpose, usually providing the best performance and energy efficiency, at the cost of flexibility and expensive design. 18, 21, 108

Cadence Cadence is a registered trademark of Cadence Design Systems, Inc. 34, 105, 161, 182

CEA-LETI Research institute for electronics and information technologies, based in Grenoble, France. 3εFERRO project partner. 21, 48, 52, 54, 55

checkpointing Action of storing the state of a circuit, program or memory to anticipate the need to restore it at a later point, in response to a power loss or other need for the previous state. It can for instance be used to store a default or “known-good” state, to accelerate backtracking or context changes during computation. 86, 92

cloud The “cloud” consists of servers rented from a third-party. Typically, these servers are powerful computers located in a datacenter, and the use of a single server can be divided among multiple users. This enables companies more flexibility with the scaling of their computing capacity and its geographical location. 16, 17, *see edge computing*

CMOS Complementary MOS, combining an **n-MOS** and a **p-MOS** stage to lower static current consumption. 3, 20, 21, 48–50, 53, 64–66, 83, 85–88, 106, 108, 110, 125, 144, *see n-MOS, p-MOS & MOS*

D flip-flop A D flip-flop is a circuit that samples its input on rising clock edges, and maintains it as its output value (until the next rising clock edge). It is widely used to create shift registers, or delay an input signal by one clock period. 95, 96

Demokritos National center for scientific research (NRCPS) “Demokritos”, Greece, Athens (Greek: Εθνικό Κέντρο Έρευνας Φυσικών Επιστημών (Ε.Κ.Ε.Φ.Ε.) « Δημόκριτος »). 3εFERRO project partner. 21

ECL Research university in greater Lyon, France (French: École centrale de Lyon). 3εFERRO project partner. 21, 92, 117

edge computing An extension of the cloud computing paradigm, edge computing designates pre-computing that happens at the periphery, or edge, of the cloud, before sending data to the cloud. That pre-computing usually saves bandwidth. 16, 17, *see cloud*

EPFL École Polytechnique Fédérale de Lausanne public research university located in Lausanne, Switzerland. 3εFERRO project partner. 5, 21, 36

fan-out Number of gate inputs able to be driven by a single logic gate output. 89

FeRAM Ferroelectric **RAM**, usually 1T1C, can be **FeFET**-based. 49, 50, *see RAM*

flash Type of non-volatile memory based on floating transistor gates, where information is stored through charges trapped on the transistor gate, which is floating due to being connected in series with a capacitor. 44, 46, 48–50, 76, 83, 85, 110, 111

FPGA Field-Programmable Gate Array: reconfigurable digital circuit commonly used for prototyping or to replace ASICs in low-volume productions. 21, 52, 82, 108, *see ASIC*

- framerate** Also called frame-rate, frames per second, images per second, display frequency: speed at which images are captured or displayed, has a direct effect on the bandwidth required for transmitting and processing video.
- full adder** Logic circuit capable of adding three binary inputs, with two typically being digits of words being added, and the third being the carry computed after adding the previous digits. See [section 4.6.3](#). [98–100](#), [102](#), [103](#), [105](#), [107](#), [108](#), [114](#), [181](#), [182](#)
- fuzzing** Fuzz-testing is a development practice that applies random data or values to function inputs in order to uncover bugs and unforeseen edge cases. [145](#)
- FZJ** Jülich Research Centre, Jülich, Germany (German: Forschungszentrum Jülich). 3εFERRO project partner. [21](#)
- GlobalFoundries** Multinational semiconductor manufacturing and design company, and NaMLab partner. [47](#), [58](#), [83](#), [125](#), [159](#)
- GPIO** General Purpose Input/Output, connectors on a microcontroller of FPGA-based platform whose behavior can be specified by the user. [108](#)
- HDMI** High Definition Media Interface, a customer media interface commonly used by digital cameras and television sets. [108](#)
- HfZrO₂** Hafnium Zirconium Oxide, whose orthorhombic phase is ferroelectric [16](#), [21](#), [22](#), [24](#), [26](#), [30](#), [35](#), [47](#), [49](#), [50](#), [53–55](#), [60](#), [63](#), [76](#), [77](#), [83](#), [85](#), [140](#), [146](#), [160](#)
- high-k** High permittivity (κ or ϵ_r) dielectrics repolarize more than low- κ materials in response to an externally applied electric field, allowing them to better guide and propagate electric fields. [18](#), [20](#), [49](#), [50](#), [159](#)
- I_{DS} Drain-Source current on a MOSFET, which depends on V_{GS} . [61](#), [70](#)
- I_{DS} — V_{GS} Classical representation of the transconductance of a MOSFET transistor, used for describing performance characteristics. [8](#), [46](#), [74](#), [75](#), [81](#), see V_{GS} & I_{DS}
- imprint** Describes asymmetric response of a ferroelectric material to electric fields of the same amplitude but opposite polarization[Mue+13b]. The P - V curve is shifted to the right or left on the voltage axis depending on imprint. [28](#), [49](#), [60](#)
- INL** Lyon Institute of Nanotechnology, Lyon, France (French: Institut des Nanotechnologies de Lyon). [5](#), [21](#), [92](#), [117](#)
- kernel** also called a mask or convolution matrix, small (usually 5×5 or less) matrix used in a two-dimension convolution with an image for processing purposes. See [section 4.6.1](#). [93–95](#), [97](#), [99](#), [100](#), [103](#), [104](#), [108](#), [179](#)
- LIFT** In-house multi-objective optimization platform, described in [Bri21]. [117](#), [139](#)
- MAD200** ferroelectric HfZrO₂ demonstrator process at CEA-LETI and STMicroelectronics based on the HCMOS9A LP/HV OxRAM process[HC MOS]. MAD stands for Memory Advanced Demonstrator. [5](#), [7](#), [11](#), [54](#), [55](#), [58](#), [63](#), [67](#), [68](#), [70](#), [71](#)
- MATLAB** Proprietary programming language and numeric computing environment developed by MathWorks. [36](#), [104](#), [117](#), [118](#), [161](#), [173](#), [176](#), [177](#)
- memory window** Distance separating multiple memory states when reading a memory. The wider the memory window, the easier it is to differentiate multiple states. If the memory window becomes too small, the memory may not function properly as it becomes impossible to discriminate memory state from sensor noise. [119](#)
- MRAM** Magnetoresistive RAM, possibly programmed using Spin-Transfer Torque (STT). Stores information in a ferromagnetic element, read via magnetic tunnel junction. [49](#), [50](#)
- n-MOS** n-channel MOSFET [59](#), [65](#), [69](#), [81](#), [82](#), [84](#), [91](#), [108](#), [159](#), see MOS
- NaMLab** Nanoelectronics Materials Laboratory gGmbH, research organization originally focused on material research for future memory devices, based in Dresden, Germany. 3εFERRO project partner. [5](#), [21](#), [37](#), [47](#), [68](#), [74](#), [92](#), [102](#), [103](#), [119](#), [125](#), [160](#)
- NAND** Complemented And logic gate, symbol $\bar{\wedge}$; a universal logic gate, and one of the most common ones. [85](#), [89](#), [90](#), [111](#), [125](#), [127](#), [128](#), [170](#)

- NIMP** National Institute of Material Physics, Bucharest, Romania (Romanian: Institutul Național de Cercetare-Dezvoltare pentru Fizica Materialelor). 3εFERRO project partner. 21
- Non-Volatile Memory** Memory that retains the information it stores without needing any external power supply. See subsection 2.3.2. 3, 7, 21, 23, 44, 45, 49, 50, 82, 92, 108, 136, 138
- NOR** Complemented OR logic gate, symbol $\bar{\vee}$; common universal logic gate. 85
- normally-off** Normally-off (computing) is a design paradigm where the computing element is expected to remain powered-off most of the time, usually to preserve energy, and perform its computation in short bursts of activity. Power-on and power-off delays are critical for this application. 3, 49, 79, 92, 93, 113, 129, 137
- OCEAN** Open Command Environment for ANalysis, Scripting language and simulation control environment used within Virtuoso® Analog Design Environment, based on the SKILL programming language. 118, see SKILL
- GNU Octave** High-level programming language for scientific computing and numerical computation, free software under the GPL license and largely MATLAB-compatible. 14, 36, 104, 173, 176, 177
- OxRAM** Oxide-based ReRAM, where the resistance of an oxide layer is modulated by creating conductive filaments. 54, 160, see ReRAM
- p-MOS** p-channel MOSFET 65, 69, 159, see MOS
- PCM** Phase-change memory, storing information as the conductivity of a material, which can be changed by altering its phase with controlled heating and crystallization. 49, 50
- pre-warming** Action of moving a copy of data inside a faster memory tier, to accelerate an anticipated need for low-latency or high-throughput access to the data normally stored in a slower memory tier. It is directly related to the concept of caching, with warm (recently accessed, up-to-date) and cold caches. 86
- P-V*** $P = f(V)$ curve, used for characterizing ferroelectric properties as described in subsection 2.1.2. 11, 14, 26–29, 32–34, 37, 38, 40, 44, 45, 160, 161, 173
- ReRAM** Resistive RAM, storing information as the resistance state. The resistance can be changed by applying electrical currents. 49, 50, 161, see resistance state & OxRAM
- resistance state** Used to describe the state of devices with variable or controlled electrical conductivity, such as transistors. Often used to distinguish between two states, *high resistance state* and *low resistance state*, but can also be used with more than two levels. 87, 90–92, 161, see RS, OxRAM & FeFET
- S-curve** *P-V* curve for a ferroelectric, displaying a single continuous, S-shaped curve instead of an hysteresis loop, as displayed in Figure 2.12. 36, 37, 39, see *P-V*
- SKILL** Lisp-based[Bar90] “interactive programming language” used throughout Cadence Design Systems environments. 14, 127, 165, 167
- Spectre** Proprietary SPICE circuit simulator with support for the Verilog-A modeling language, owned and distributed by Cadence Design Systems. 34, 105, 144
- STMicroelectronics** Franco-Italian multinational electronics and semiconductors manufacturer. 3εFERRO project partner. 21, 50, 54, 58, 160
- SystemC** System-level simulation and architecture description language based on the C++ language. 130, 131, 140
- Verilog** RTL Hardware Description Language (HDL) standardized as IEEE 1364. Is used to describe electronic systems for simulation and logic synthesis. 14, 99, 104, 105, 177–179
- Verilog-A** Continuous-time modeling language, derivative of the Verilog language. 14, 43, 105, 161, 171, 172, see Verilog
- V_{GS} Potential between gate and source on a MOSFET, also usually referred to as the gate voltage. 160
- XOR** Exclusive OR logic gate, symbol \oplus or \veebar ; most complex case on Karnaugh’s table, and often the most transistor-intensive logic gate. 90, 91

Acronyms

- ADC** Analog-to-Digital Converter 58, 123
AFM Atomic Force Microscopy 27
ALD Atomic Layer Deposition 22
ALU Arithmetic and Logic Unit 131
- BEoL** Back-End of Line 3, 11, 46–48, 50, 53–55, 58–60, 77, 143, 144
BL Bit Line 55–57, 69, 71, 72, 74, 83–85, 119–121, 123
- CAM** Content-Addressable Memory 48, 66, 68, 77
CGRA Coarse-Grained Reconfigurable Array 52
CNN Convolutional Neural Network 93, 145
CPU Central Processing Unit 18, 19, 21, 130–132, 140, 145
- DK** Design Kit 71, 72
DRAM Dynamic Random Access Memory 20, 44, 45, 54, 55, 57, 68, 70, 71, 77, 85, 86, 110, 119, 123, 129, 131–133, 143
DRC Design Rules Check 72
DSE Design-Space Exploration 3, 49, 77, 113, 114, 116, 117, 119, 121–128, 139–141, 144–146
DSP Digital Signal Processor 93
DTCO Design-Technology Co-Optimization 113, 135, 140, 141
- E_C Coercive Electric field 23–27, 29, 30, 44–47, 55, 80, 83, 146
- FDSOI** Fully Depleted Silicon on Insulator 16, 21
FeCap Ferroelectric Capacitor 22, 30, 44, 46, 48, 53–55, 57–60, 62, 65–77, 85, 119, 121–123, 125, 132, 143–146, 184
FeFET Ferroelectric Field-Effect Transistor 3, 11, 13, 16, 22, 27, 30, 36, 43–51, 58–60, 62, 63, 67–72, 76, 77, 79–94, 98, 99, 102, 103, 106–108, 110, 113, 119, 125, 127–129, 133, 138, 143–146, 159, 170, 181, 182
FEoL Front-End of Line 11, 46, 50, 54, 58, 60, 64, 76
FET Field Effect Transistor 19, 45, 46, 80, 86, 89
FGMOS Floating-Gate MOSFET 48, 85, 110, 144
FinFET Fin FET 16, 19
FIR Finite Impulse Response 93
FORC First Order Reversal Curve 40
FPS Frames Per Second 98, 108, *see* framerate
FTJ Ferroelectric Tunnel Junction 30, 40, 46, 49, 68, 70, 71, 77
- GAAFET** Gate-All-Around FET 19
GPU Graphics Processing Unit 21
- HDL** Hardware Description Language 161
- I/O** Input/Output 19, 20, 97, 100, 108
IMC In-Memory Computing 67, 129
IoT Internet of Things 16, 17, 21, 49
IPC Inter-Process Communication 11, 118, 125, 139, 144, 169, 170
IRDS International Roadmap for Devices and Systems 19
ITRS International Technology Roadmap for Semiconductors 19
- LiM** Logic-in-Memory 12, 59, 79, 92, 110, 113, 114, 129–132, 136–138, 140, 143–145

- LSB** Least-Significant Bit 108
- LUT** lookup table 12, 133, 137, 138
- MBD** Molecular Beam Deposition 22
- MCAM** Multi-bit Content-Addressable Memory 48
- MFM** Metal-Ferroelectric-Metal 46, 47, 58, 59, 76
- MFS** Metal-Ferroelectric-Semiconductor 46, 47
- ML** Match Line 66, 68, 71
- MLC** Multi-Level Cell 26, 27, 37, 57, 58, 71, 75, 82, 119, 123, 143, 146
- MOS** Metal-Oxide-Semiconductor 45–47, 65, 80, 82, 144
- MOSFET** Metal-Oxide-Semiconductor Field-Effect Transistor 17, 18, 30, 48, 61, 62, 73, 83, 86, 160, 161, 163
- MSB** Most-Significant Bit 108
- NCFET** Negative-Capacitance Field-Effect Transistor 22, 37, 45, 49
- NWB** Non Write Back 12, 131, 132, 138, 139
- PL** Plate Line 55–57, 64, 66, 68, 69, 71, 72, 74, 83–85, 120, 121
- PLD** Pulsed Laser Deposition 22
- Pr** remanent polarization 11, 27, 28, 48, 57, 58, 60, 61, 75, 120–123, 182
- PsFeFET** Pseudo-FeFET 11, 13, 46, 54, 58–65, 70, 71, 73, 76, 77, 87, 92, 110, 143–145
- PUND** Positive-Up, Negative-Down – Specific waveform used for characterizing ferroelectric capacitors, described in subsection 2.1.4. 11, 27, 30, 32–34, 36, 38, 173
- PVD** Physical Vapor Deposition 22
- RAM** Random-Access Memory 48, 76, 159–161
- ROI** region of interest 36–38
- RS** Resistance State 89, 90, *see* resistance state
- RTL** Register Transfer Level 103, 131, 161
- SRAM** Static Random Access Memory 20, 44, 50, 91, 92, 110
- SSD** Solid-State Disk 44, 49
- TCAD** Technology Computer-Aided Design 49
- TCAM** Ternary Content-Addressable Memory 13, 48, 54, 66–68, 71, 77, 83, 143
- TPU** Tensor Processing Unit 21
- V_C Coercive Voltage 11, 24, 27, 28, 31, 34, 43, 44, 47, 55, 57, 64–66, 69, 74, 80, 84–86, 89, 102, 105, 106, 121, 123, 146
- VO** Voltage Output 89, 90
- V_{th} threshold voltage 8, 11, 30, 65, 72, 79–83, 85, 86, 89, 106–108, 127, 144
- WB** Write Back 12, 50, 77, 131, 132, 135, 138, 146
- WL** Word Line 55, 56, 68, 69, 72, 74, 83, 84, 120, 123
- WORM** Write Once, Read Many 146

Appendix A

Code listings

LISTING A.1: **SKILL** Metric extraction from waveforms for 1T1C bitcell

```

1  analysis( 'tran ?stop "800u" ?method "gear2" )
2  desVar( "low_noise_option" 1 )
3  desVar( "L" tL )
4  desVar( "W" tW )
5  desVar( "atot_fe" Ac )
6  desVar( "vwl" vwl )
7  desVar( "vprog" vprog )
8  desVar( "vread" vread )
9
10 ; slewr in s/V
11 desVar( "tfallprogBL" (slewr * vprog) )
12 desVar( "triseprogBL" (slewr * vprog) )
13 desVar( "tfallreadBL" (slewr * vread) )
14 desVar( "trisereadBL" (slewr * vread) )
15 desVar( "tfallprogPL" (slewr * vprog) )
16 desVar( "triseprogPL" (slewr * vprog) )
17 desVar( "tfallWL" (slewr * vwl) )
18 desVar( "triseWL" (slewr * vwl) )
19
20
21 envOption(
22     'cmd64bit t
23     'analysisOrder list("tran")
24 )
25 save( 'i "/N0/D" )
26 temp( 27 )
27 run()
28 selectResult( 'tran )
29
30
31 (let
32     (
33         (Vc 1.2) ; For now, look at BL-PL, later look at n
34         (P_switch 100mv) ; Internal polarization threshold to measure
35         switching time.
36         (Vfe ((v "/BL" ?result "tran") - (v "/PL" ?result "tran"))) ; BL-PL
37         (Vp (v "/pint" ?result "tran")) ; Pint
38
39         ; First: write 1<-0
40         (tb_w10 420us) ; time begin write 0 over 1
41         (te_w10 444us) ; time end ; Those two declarations might become
42         redundant (unfortunately, no "let*" variant)
43         (iw10 (clip (i "/N0/D" ?result "tran") 420us 444us)) ; current
44         waveform for above region
45         ew_10 ; energy for writing a 0 over a 1, assumes clip works
46         ipk_w10 ; peak current for writing a 0 over a 1 (relies on clip)
47
48         ; read with 1 after write 0

```

```

46 (tb_r1_w0 520us) ; read with a 1 after writing 0
47 (te_r1_w0 544us)
48 (i_r1_w0 (clip (i "/N0/D" ?result "tran") 520us 544us))
49 er1_w0
50 ipk_r1_w0
51
52 ; write with 1 after reading (writing) with 1
53 (tb_w11 620us)
54 (te_w11 644us)
55 (iw11 (clip (i "/N0/D" ?result "tran") 620us 644us))
56 ew_11
57 ipk_w11
58
59 ; read with 1 after writing with 1
60 (tb_r1_w1 720us) ; read with a 1 after writing 1
61 (te_r1_w1 744us)
62 (i_r1_w1 (clip (i "/N0/D" ?result "tran") 720us 744us))
63 er1_w1
64 ipk_r1_w1
65
66 P_win ; max p. int difference , giving a window
67 tw_11 ; time to write 1 from 1 TODO
68 tw_10
69 tw_01
70 )
71
72 ;;
73 ew_10 = (integ ((abs iw10) * (abs Vfe)))
74 ipk_w10 = (ymax -iw10)
75 ;;
76 er1_w0 = (integ ((abs i_r1_w0) * (abs Vfe)))
77 ipk_r1_w0 = (ymax i_r1_w0)
78 ;;
79 ew_11 = (integ ((abs iw11) * (abs Vfe)))
80 ipk_w11 = (ymax iw11)
81 ;;
82 er1_w1 = (integ ((abs i_r1_w1) * (abs Vfe)))
83 ipk_r1_w1 = (ymax i_r1_w1)
84
85 tw_10 = (delay
86   ?wf1 Vfe ?value1 -Vc ?edge1 'falling ?td1 tb_w10
87   ?wf2 Vp ?value2 -P_switch ?edge2 'falling ?td2 0 ?stop te_w10)
88
89 tw_01 = (delay
90   ?wf1 Vfe ?value1 Vc ?edge1 'rising ?td1 tb_r1_w0
91   ?wf2 Vp ?value2 P_switch ?edge2 'rising ?td2 0 ?stop te_r1_w0)
92
93 ;;
94 chrg_win = (value
95   iinteg( (clip (i "/N0/D" ?result "tran") 415us 444us) )
96   444u) ; note: WL still active
97
98 restable["ipk_r1_w0"] = ipk_r1_w0
99 restable["ipk_r1_w1"] = ipk_r1_w1
100 restable["ew_10"] = ew_10
101 restable["er1_w0"] = er1_w0
102 restable["er1_w1"] = er1_w1
103 restable["tw_10"] = tw_10
104 restable["tw_01"] = tw_01
105 restable["chrg_win"] = chrg_win
106
107 )
108

```

109 (inl_ipcWriteTable restable)

LISTING A.2: **SKILL** Metric extraction from waveforms for FeFET-based non-volatile **NAND** gate, as described in subsection 5.3.2

```

1 tpulse=20u
2 analysis('tran ?stop "600u" )
3 desVar( "tpulse" tpulse )
4 desVar( "trise" 0.2u )
5 desVar( "VDC" 1.5 )
6 desVar( "VPROG" 5 )
7 desVar( "Wfe" Wfe ) ;500n, inserted above automatically
8 desVar( "Lfe" Lfe ) ;500n, ditto
9 envOption(
10 'cmd64bit t 'userCmdLineOption "+lite" 'analysisOrder list("tran") )
11 save( 'i "/V3_vdd/PLUS" )
12 save( 'v "/vclk" "/pint" "vout" )
13 temp( 27 )
14 run()
15 selectResult( 'tran )
16
17 (let
18 (
19 (Vc 1.2) ; For now, look at BL-PL, later look at n
20 (Vt 0.7) ; threshold voltage, true
21 (Vtn 0.4) ; threshold voltage, false
22 (P_switch 100m) ; Internal polarization threshold to measure sw time
23 (Vp (v "/pint" ?result "tran")) ; Internal polarization terminal
24 (Vo (v "/vout" ?result "tran"))
25 (Vclk (v "/vclk" ?result "tran"))
26
27 ; Write 0
28 (tw_1 480u)
29
30 ; Write 1
31 (tw_0 340u)
32
33 ; 11
34 (ttt 400u) ott_valid vtt iprch
35
36 ; 00
37 (tff 560u) off_valid vff
38
39 ; 10
40 (ttf 440u) otf_valid vtf
41
42 ; 01
43 (tft 520u) oft_valid vft min_av
44 )
45
46 iprch = (clip (i "/V3_vdd/PLUS" ?result "tran")
47 (ttt + tpulse + tpulse / 2 )
48 (ttt + tpulse + tpulse + tpulse / 2))
49 eprch = (integ ((abs iprch) * (abs Vclk)))
50
51 vtt = (value Vo (ttt + tpulse / 2 ))
52 vff = (value Vo (tff + tpulse / 2 ))
53 vtf = (value Vo (ttf + tpulse / 2 ))
54 vft = (value Vo (tft + tpulse / 2 ))
55
56 ott_valid = ( vtt < Vt)
57 off_valid = ( vff > Vt)
58 otf_valid = ( vtf > Vt)

```

```
59     oft_valid = ( vft > Vt)
60
61     ; send valid=1 if every check is valid. Needs to be fp for the IPC
62     restable["valid"] = (if (and ott_valid off_valid otf_valid oft_valid)
63         1.0 0.0)
64     restable["vtt"] = vtt
65     restable["vff"] = vff
66     restable["vtf"] = vtf
67     restable["vft"] = vft
68     restable["eprch"] = eprch
69 )
70
71 (inl_ipcWriteTable restable)
```

LISTING A.3: 1T1C Design space exploration python script as described in subsection 5.3.1, leveraging the IPC described in subsection 5.2.2.

```

1  #!/usr/bin/env python3
2
3  import sys
4  import json
5
6  sys.path.append("@cadenceipc/")
7
8  from ipc import oceanConnector
9
10 def sim_iteration(loopbody, parameters):
11     # Construct params str
12     pstr = ""
13     for key in parameters:
14         pstr = pstr + f"{key}={parameters[key]:0.10g}\n"
15     print("Sending params", parameters)
16     i.sendAndWaitCommand(pstr)
17     print("Sending loop")
18     i.sendAndWaitCommand(loop, 10*60) # Timeout 10 minutes
19     res = i.getTable()
20     print("Got results ", res)
21     return res
22
23
24 i = oceanConnector(["newoceanmad", "--nograph"], "/tmpfolder")
25 i.connect()
26
27 print("Sending preamble")
28 with open('preamble.ocn') as pre:
29     i.sendAndWaitCommand(pre.read())
30
31 with open('loop.ocn') as lfile:
32     loop = lfile.read()
33
34 simFamily = []
35 for tW in [130e-9, 140e-9, 160e-9, 180e-9]:
36     for tL in [150e-9, 170e-9, 200e-9]:
37         for sqrtAc in [100e-9, 150e-9, 180e-9, 300e-9]:
38
39             simFamily.append({"parameters": {
40                 "tW": tW,
41                 "tL": tL,
42                 "Ac": sqrtAc**2,
43                 "slewr": 5e-9, # 5ns/V
44                 "vprog": 3.6,
45                 "vread": 3.6,
46                 "vwI": 3.6
47             }})
48
49
50 for sim in simFamily:
51     sim["result"] = sim_iteration(loop, sim["parameters"])
52
53 with open("results_explo.json", "w") as resfile:
54     json.dump(simFamily, resfile)

```

LISTING A.4: FeFET-based non-volatile NAND gate design space exploration script as described in subsection 5.3.2, using the IPC described in subsection 5.2.2. loop.ocn corresponds to Listing A.2.

```

1  #!/usr/bin/env python3
2
3  import csv
4  from itertools import product
5  import sys
6  import os
7
8  sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))
9
10 from ipc import oceanConnector
11
12 i = oceanConnector(["ocean", "--nograph"],
13                   "/tmp/runtimeoceanfolder")
14
15 i.connect()
16
17 with open('preamble.ocn') as pre:
18     i.sendAndWaitCommand(pre.read())
19
20 with open('loop.ocn') as lfile:
21     loop = lfile.read()
22
23 tW = [1.2e-6, 1.3e-6, 1.4e-6, 1.5e-6]
24
25 tL = [300e-9, 380e-9, 500e-9, 560e-9, 650e-9, 780e-9, 880e-9, 1000e-9,
26       1.2e-6, 1.4e-6]
27
28 combinations = product(tW, tL)
29
30 with open('results.csv', 'a') as csvfile:
31     writer = csv.writer(csvfile)
32     # writer.writerow(['Wfe', 'Lfe', 'valid', 'vtt', 'vff', 'vtf', 'vft', 'eprch'])
33     # The above line is the csv header. Simulate each parameter set:
34     for params in combinations:
35         print(f'trying {params}')
36         ctext = f'Wfe={params[0]:0.10g}\nLfe={params[1]:0.10g}\n'
37         i.sendcommand(ctext)
38         i.sendAndWaitCommand(loop, 10*60)
39         t = i.getTable()
40         writer.writerow([
41             params[0], params[1],
42             t["valid"],
43             t["vtt"], t["vff"], t["vtf"], t["vft"],
44             t["eprch"]
45         ])

```

LISTING A.5: Verilog-A Data Serializer

```

1  `include "constants.vams"
2  `include "disciplines.vams"
3
4  module va_save_8b(clk, data);
5  input clk;
6  electrical clk;
7  input [7:0] data;
8  electrical [7:0] data;
9
10 integer fd_out_data;
11 integer bin_out, i;
12
13 parameter real vth = 0.3;
14 parameter string out_filename = "save8.bin";
15
16 analog begin
17
18     @(initial_step) begin: openfile
19         fd_out_data = $fopen(out_filename, "w"); // WB
20         if (fd_out_data == 0) begin
21             $display("Could not open file");
22             $finish;
23         end
24     end
25
26     @(final_step)
27         $fclose(fd_out_data);
28
29     @(cross ( V(clk)-vth, 1)) begin: serialize
30         begin
31             bin_out = 0;
32             generate i(7,0)
33                 bin_out = bin_out + ((V(data[i]) > vth) << i);
34             $display("read %h", bin_out);
35             $fwrite(fd_out_data, "%c", bin_out);
36         end
37     end
38
39 end // analog
40
41 endmodule

```

LISTING A.6: Verilog-A Data Serializer with enable signal

```

1  `include "constants.vams"
2  `include "disciplines.vams"
3
4  module va_save_out_data(clk, out_data, out_data_valid);
5  input clk;
6  electrical clk;
7  input [7:0] out_data;
8  electrical [7:0] out_data;
9  input out_data_valid;
10 electrical out_data_valid;
11
12 integer fd_out_data;
13 integer bin_out, i;
14
15 parameter real vth = 0.3;
16 parameter string out_filename = "filter_output.bin";
17
18 analog begin
19
20     @(initial_step) begin: openfile
21         fd_out_data = $fopen(out_filename, "w"); // WB
22         if (fd_out_data == 0) begin
23             $display("Could not open file");
24             $finish;
25         end
26     end
27
28     @(final_step)
29         $fclose(fd_out_data);
30
31     @(cross ( V(clk)-vth, 1)) begin: serialize
32         if (V(out_data_valid) > vth)
33             begin
34                 bin_out = 0;
35                 generate i(7,0)
36                     bin_out = bin_out + ((V(out_data[i]) > vth) << i);
37                 $display("read %h", bin_out);
38                 $fwrite(fd_out_data, "%c", bin_out);
39             end
40         end
41
42 end // analog
43
44 endmodule

```

LISTING A.7: GNU Octave (MATLAB-compatible) code for fitting Landau coefficients to experimental PUND-processed P - V curves

```

1 clear all
2
3 % Dataset-specific data loading
4 % Note that the code could be simplified if the dataset was prepared by
5 % splitting the hysteresis code in a top and bottom half.
6 fields = {'V+' [V]', 'P1 [uC/cm2]'};
7
8 load dataset;
9 x = dataset.(fields{1}); % Voltage values
10 y = dataset.(fields{2}); % Polarization values in  $\mu\text{Ccm}^{-2}$ 
11 tfe = dataset.( 'Thickness [nm]')*1e-9;
12
13 % Plot raw data
14
15 % Legends
16 prettyfields = {'V (V)', 'P ( $\mu\text{C}\cdot\text{cm}^{-2}$ )'};
17 latexfields = {'V (\unit{volt})', ...
18 'P (\unit{microcoulomb\per\square\centi\meter})'}
19
20 figure(1); clf; hold on;
21 title('Identification of Regions of interest for fitting experimental data')
22 plot(x,y, 'displayname','Experimental data');
23 legend show; legend location southeast
24
25 xlabel(prettyfields{1}); ylabel(prettyfields{2});
26
27 % Look for points of interest
28 % Specifically, both ends of the hysteresis, and the furthest points from the
29 % line that passes trough both
30
31 values = [x'; y'];
32 [value1, pos1] = min(values');
33 [value2, pos2] = max(values');
34 extrema = [pos1; pos2]'; % These are the furthest points from the center
35
36 % We choose to take the min and max X, trace a line between them
37 % (the 'median line'), take the furthest points from it on each side,
38 % and start grouping closest points together
39
40 start_hyst = extrema(1,1);
41 x1 = [x(start_hyst); y(start_hyst)];
42 stop_hyst = extrema(1,2);
43 x2 = [x(stop_hyst); y(stop_hyst)];
44
45 % x1 and x2 are the start and stop points of the curve when looking at x values
46 plot([x1(1),x2(1)],[x1(2),x2(2)], 'o', 'displayname', 'Extrema')
47
48 plot([x(start_hyst),x(stop_hyst)], [y(start_hyst), y(stop_hyst)], ...
49 'displayname','median split'); % Plot the median, or bisector line
50
51 diagonal_slope = (y(stop_hyst)-y(start_hyst))/(x(stop_hyst)-x(start_hyst));
52 diagonal_y_at_origin = y(start_hyst) - diagonal_slope*x(start_hyst); % 0 usual.
53 y_on_diagonal = diagonal_slope*x + diagonal_y_at_origin;
54
55 % note: does not work if the top/bottom curves don't wrap,
56 % and edge points could belong both to the top and bottom part
57 a = min(start_hyst, stop_hyst); b = max(start_hyst, stop_hyst);
58 part_direct = a:b;
59 part_wrap = [b:size(values, 2), 1:a];
60
61 if sum(y(part_direct) > y_on_diagonal(part_direct)) > ...

```

```

62     sum(y(part_wrap) > y_on_diagonal(part_wrap))
63     part_top = part_direct; % More points above the diagonal
64     part_bottom = part_wrap;
65 else
66     part_top = part_wrap;
67     part_bottom = part_direct;
68 end
69
70 % Compute the orthogonal projection on the diagonal for every point of the loop
71 % This is to find the point furthest from the median split
72 % See Listing A.8 for the code.
73
74 ortho = orthoproj([x,y],
75                  [x(stop_hyst)-x(start_hyst), y(stop_hyst)-y(start_hyst)],...
76                  [x(start_hyst), y(start_hyst)]);
77
78 sqdist2proj = ortho-values';
79 sqdist2proj = sqdist2proj(:,1).^2 + sqdist2proj(:,2).^2;
80
81 sqdist2proj_top = sqdist2proj; sqdist2proj_top(part_top) = 0;
82 [~, pos_furthest_top] = max(sqdist2proj_top);
83
84 sqdist2proj_bot = sqdist2proj; sqdist2proj_bot(part_bottom) = 0;
85 [~, pos_furthest_bot] = max(sqdist2proj_bot);
86
87 u = [x(pos_furthest_bot); x(pos_furthest_top)];
88 v = [y(pos_furthest_bot); y(pos_furthest_top)];
89
90 plot(u,v, '+', 'displayname', 'Furtherst')
91
92 %%%%%%%%%% Use the POIs to compute regions of interest %%%%%%%%%%
93 % Regions of interest are the regions that are not discarded to compute
94 % the s-shaped curve
95 %[u,v]=ginput(2); % We can ask the user to manually select POIs with this
96
97 % The following two lines are only useful if manually entering points, to select
98 % The points on the curve closest to the user-selected regions (norm 2)
99 [~, pos_user1] = min( ((x-u(1)).^2 + (y-v(1)).^2));
100 [~, pos_user2] = min( ((x-u(2)).^2 + (y-v(2)).^2));
101
102 a = min(stop_hyst, pos_user1); % just make sure indexes a<b for the calculations
103 b = max(stop_hyst, pos_user1);
104
105 direct = a:b;
106 modulo = [a:size(values, 2), 1:b];
107 if abs(min(direct) - max(direct)) < abs(min(modulo) - max(modulo))
108     range1 = direct;
109 else
110     range1 = modulo; % Take the shortest (x-wise) path
111 end
112
113 a = max(start_hyst, pos_user2);
114 b = min(stop_hyst, pos_user2);
115
116 direct = a:b;
117 modulo = [a:size(values, 2), 1:b];
118 if abs(min(direct) - max(direct)) < abs(min(modulo) - max(modulo))
119     range2 = direct;
120 else
121     range2 = modulo; % Take the shortest (x-wise) path
122 end
123 x_partial = [x(range1); x(range2)];
124 y_partial = [y(range1); y(range2)];

```

```

125
126
127 plot(x(range2),y(range2), '—', 'linewidth',3, 'displayname',...
128      'ROI 2', 'color', [.64 .37 .51])
129 plot(x(range1),y(range1), '—', 'linewidth',3, 'displayname', 'ROI 1')
130 set(gca,'linewidth',2.5,'FontSize',22,'ticklength',[0.025 0.025],...
131      'PlotBoxAspectRatio',[1 0.85 1]);
132
133 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Polynomial fit %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
134 n=6; % fit degree
135 % Odd numbers up to n, polynomial coefficients to keep
136 odds = find(mod(1:n,2) ~= 0);
137 p = polyfitc(y_partial,x_partial, odds); % Odd polynomial fit, see Listing A.9
138
139 % Note how the polynomial fit is performed on flipped axes.
140 % boundary Y values are taken as a new X axis, and the plot is inverted to
141 % compensate.
142 X_sim_flip = linspace(y(start_hyst), y(stop_hyst), 1000);
143 Y_sim_flip = polyval(p, X_sim_flip);
144
145 polysign = {'','+'}; % to avoid displaying 2x +3x^3 etc (see +-)
146 polysign = {polysign{1+(p>0)}};
147 polyexp = sprintf('%gP%s%gP^3%s%gP^5',p(5),polysign{3},p(3),polysign{1},p(1))
148
149 % Compute Landau coefficients
150 % -> V = 2·tfe·p + 4·tfe·p3+6·tfe·p
151 pconv = 1e-2; % 1μC cm-2 = 10-2 coulomb/m2
152 alpha = p(5)/(2*tfe*pconv^1)
153 beta = p(3)/(4*tfe*pconv^3)
154 gamma = p(1)/(6*tfe*pconv^5)
155
156 fprintf('Polynomial coefficients:\nV=%s\nalpha=%d, beta=%d, gamma=%d\n',...
157        polyexp, alpha, beta, gamma);
158 xlabel(prettyfields{1}); ylabel(prettyfields{2});
159
160
161 % Plot Final Fit
162 figure(2); clf; hold on;
163 plot(x,y, 'displayname', 'Experimental data')
164 plot(Y_sim_flip, X_sim_flip, 'displayname', 'Polynomial fit');
165 title('Fit to experimental data')
166 legend show
167 legend location southeast
168 set(gca,'linewidth',2.5,'FontSize',22,'ticklength',[0.025 0.025],...
169      'PlotBoxAspectRatio',[1 0.85 1]);

```

LISTING A.8: GNU Octave (MATLAB-compatible) code for orthogonal projection, dependency of Listing A.7.

```

1 % Orthogonally projects a list of points [x,y] (size n,2)
2 % On a line defined by a unitary vector [x,y] and a passing point [x,y].
3 % Returns the coordinates of every point.
4 function ortho = orthoproj(points, unitary, passing)
5     unitary = unitary/norm(unitary); % Ensure
6
7     OB = passing;
8     BA = -OB + points; % Points are OA: BA = BO + OA = -OB + OA
9     BH = (BA(:,1) .* unitary(1) + BA(:,2) .* unitary(2)) * unitary; % Dot product
10    OH = OB + BH;
11
12    ortho = OH;
13 end

```

LISTING A.9: GNU Octave (MATLAB-compatible) code for a polynomial fit with select coefficients constrained to 0. Adapted from[J14], dependency of Listing A.7.

```

1 function p=polyfitc(x,y,nvec)
2 %% from https://mathworks.com/matlabcentral/answers/
3 %     123072-curve-fitting-tool-with-custom-equation-odd-power-polynomial
4 %
5 % Simple 1D polynomial fitting with particular coefficients constrained to zero
6 %
7 %     p=polyfitc(x,y,nvec)
8 %
9 %in:
10 %
11 % nvec: A vector of integer exponents present in the polynomial.
12 % x: x data
13 % y: y data
14 %
15 %out:
16 %
17 % p: vector of polynomial coefficients in decreasing order from max(nvec) to 0
18
19 A=bsxfun(@power,x(:),nvec(:).');
20 [QQ,RR]=qr(A,0);
21 coeffs = RR\(QQ'*y(:));
22
23 p=zeros(1,max(nvec)+1);
24 p(nvec+1)=coeffs;
25 p(end:-1:1);

```

LISTING A.10: GNU Octave (MATLAB-compatible) code displaying the use of convolutional filters, used to generate Figure 4.3

```

1 clear variables; figure(1); clf;
2 original = imread('MAD200-microscope.jpg');
3 original = imresize (original, [300 356]); % Downsize for more obvious effect
4
5 %% Filter generation
6 identity = [1]; % One-by-one kernel with factor 1 as the identity
7 gauss = int8(fspecial('gaussian', [5 5], 1)*128);
8 sobel = int8(fspecial('sobel'));
9 sharp = int8(fspecial('unsharp', 0));
10
11 map = { % kernel, factor, name
12     identity, 1, 'Original';
13     gauss, 128, 'Blurred';
14     sobel, 1, 'Sobel';
15     sharp, 1, 'Sharpened'
16 };
17
18 %% Filtering and plotting
19 for i=1:size(map,1)
20     subplot(2,2,i);
21     kernel = map{i,1}; factor = map{i,2}; name = map{i,3};
22     filtered = uint8(conv2(original, kernel, 'valid')/factor); % Filter and scale
23     imshow(filtered); title(name);
24     imwrite(filtered, horzcat(name, '.jpg'));
25 end

```

LISTING A.11: Verilog description of shift register with alternate input as described in section 4.6.2. This version both samples its input, and changes its output on falling clock edges, leading to possible issues with cascaded registers if clock signals do not reach shift registers simultaneously.

```

1 /* Muxed flip-flops, otherwise known as "SR" in the synoptic document.
2  *
3  * This circuit selects between two input busses each 8 bits wide, and acts as
4  * a D-flip-flop that holds the selected value on positive clock edges.
5  */
6
7 module muxed_ff(clk, A, B, select_B, O);
8
9     parameter bus_size=8;
10     input wire [bus_size-1:0] A,B;
11     input wire select_B, clk;
12
13     output reg [bus_size-1:0] O;
14
15
16     always @(negedge clk)
17         O <= select_B ? B: A;
18
19 endmodule

```

LISTING A.12: Verilog description of shift register with alternate input as described in [section 4.6.2](#). This version samples input signal on falling edges, and updates output signals on rising edges, allowing every cascaded register to sample its input before changing the output.

```

1  /* Muxed flip-flops, otherwise known as "SR" in the synoptic document.
2  *
3  * This variant is sensitive to both edges of the clock signal:
4  * The circuit still selects between two input busses each 8 bits wide,
5  * with the "select_B" signal, but samples the selected bus on positive clock
6  * edges, and changes its output on negative edges.
7  *
8  * This is done to mitigate possible inconsistent delays in the clock tree.
9  *
10 */
11
12 module muxed_ff_2edges(clk, A, B, select_B, O);
13
14     parameter bus_size=8;
15     input wire [bus_size-1:0] A,B;
16     input wire select_B, clk;
17
18     output reg [bus_size-1:0] O;
19     reg [bus_size-1:0] sampler;
20
21
22     always @(posedge clk)
23         sampler <= select_B ? B : A;
24
25     always @(negedge clk)
26         O <= sampler;
27
28 endmodule

```

LISTING A.13: Synthesized Verilog description of a 1-bit shift version of [Listing A.12](#), as described in [section 4.6.2](#). This 1-bit variant was less complex to implement, and could be parallelized 8 times to provide an 8-bit implementation. The positive edge flip-flop was implemented with a negative edge flip-flop with an inverted clock signal as shown in [Circuit 4.14](#), reducing the implementation to three different layouts.

```

1  //////////////////////////////////////
2  // Created by: Synopsys DC Expert(TM) in wire load mode
3  // Version    : O-2018.06-SP3
4  // Date       : Fri Mar 20 09:10:19 2020
5  //////////////////////////////////////
6
7
8  module muxed_ff_2edges ( clk, A, B, select_B, O );
9      input [0:0] A;
10     input [0:0] B;
11     output [0:0] O;
12     input clk, select_B;
13     wire \sampler[0], n1, n4;
14
15     DFFPOSX1 \sampler_reg[0] ( .D(n1), .CLK(clk), .Q(\sampler[0]) );
16     DFFNEGX1 \O_reg[0] ( .D(\sampler[0]), .CLK(clk), .Q(O[0]) );
17     INVX1 U6 ( .A(n4), .Y(n1) );
18     MUX2X1 U7 ( .B(A[0]), .A(B[0]), .S(select_B), .Y(n4) );
19 endmodule

```

LISTING A.14: Excerpt of the Verilog testbench that initially feeds the kernel data to the multiplier circuit.

```

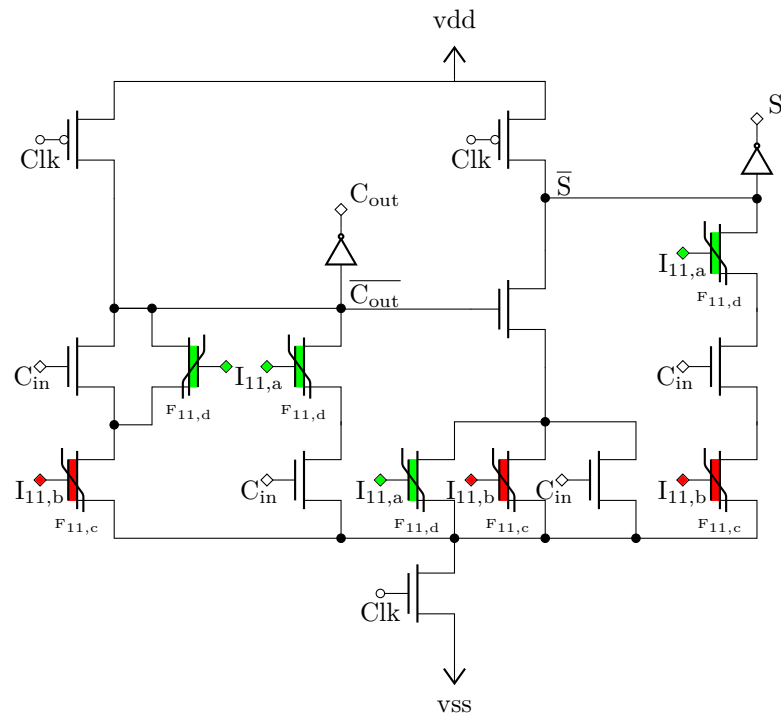
1 task feed_kernel_data(input reg only_sign_ext);
2 begin: feed
3     integer i, clk_per_mult;
4     reg [7:0] data;
5     clk_per_mult = only_sign_ext? sr_cycles_per_multiplier : 1;
6     for(i=0; i < kernel_width*kernel_width; i = i+1)
7         begin
8             // We need to feed the kernel head first to have the right
9             // values pass by first.
10            data = kernel_data[24-i];
11            in_data = only_sign_ext? {8{data[7]}} : data;
12            // *2 because the loop only has one edge
13            for ( i_sr = 0; i_sr< clk_per_mult*2; i_sr=i_sr+1)
14                begin
15                    #(serial_SR_clk_period/2) clk1 = !clk1; clk2 = !clk2;
16                end
17            end
18        end
19    endtask
20
21
22 /* Load filter coefficients trough the "debug" circuit.
23 *
24 * The goal is to have each multiplier coefficient line up with its kernel_in
25 * input. Note that we feed the kernel coefficients in the "correct" order,
26 * since the SR chain is a fifo, and the first in are the furthest in the
27 * image. However, since the image columns are going to be fed from multiplier
28 * [0,5,10,15,20] to [4,9,14,19,24] in that order, the pixels in the leftmost
29 * column of the kernel need to be burned into the last column.
30 */
31 task load_kernel;
32 begin
33     debug_enabled = 1; // Filter programming happens in debug mode
34     /*
35     * First feed the 25 coefficient trough the debug circuit, but there is
36     * a catch: each needs to be fed sr_cycles_per_multiplier (normally 8),
37     * as that is the space between two multipliers. Those coefficients
38     * are going to be stored in the second half of the multipliers, so
39     * they just consist of the sign extension bits, repeated.
40     */
41     begin: feed_kernel
42         // due to the input topology, loop in that order:
43         feed_kernel_data(1); // Only feed the sign extension bits
44         $display("Finished feeding the sign extension values");
45         feed_kernel_data(0);
46     end
47
48     /* Now that kernel has been feed, we need to align it with the
49     * multipliers by waiting for the initial latency
50     */
51     begin: fast_forward_kernel
52         in_data = 8'hxx;
53         for ( i_sr = 0; i_sr< sr_to_multiplier_latency; i_sr=i_sr+1)
54             begin
55                 #(serial_SR_clk_period/2) clk2 = !clk2; clk1 = !clk1;
56                 #(serial_SR_clk_period/2) clk2 = !clk2; clk1 = !clk1;
57             end
58         $display("finished catching up to latency, about to write");
59     end
60     vdd3_on = 1;
61     write_enable = 1;

```

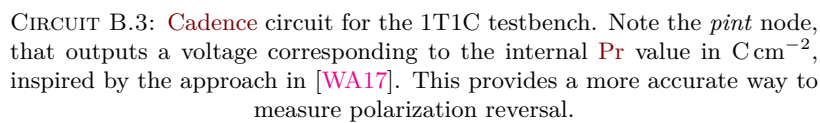
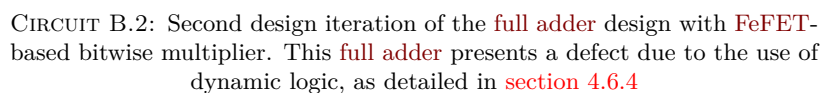
```
62     #(kernel_write_time)
63     write_enable = 0;
64     debug_enabled = 0;
65     vdd3_on = 0;
66     #2
67     $display("Wrote kernel");
68 end
69 endtask
```

Appendix B

Additional circuits



CIRCUIT B.1: First proposed version of the **full adder** with **FeFET**-based bitwise multipliers. Note that this design performs two multiplications, unlike the subsequent ones.



Appendix C

Additional tables

300 71×10^{-15}	400 126×10^{-15}	550 238×10^{-15}	550 238×10^{-15}	←Capacitor Diameter (nm) ←↓Equivalent area (m^2)
				0
X				71×10^{-15}
	X			126×10^{-15}
X	X			196×10^{-15}
			X	238×10^{-15}
X			X	308×10^{-15}
	X		X	363×10^{-15}
X	X		X	434×10^{-15}
		X	X	475×10^{-15}
X		X	X	546×10^{-15}
	X	X	X	601×10^{-15}
X	X	X	X	672×10^{-15}

TABLE C.1: Possible 2T4C combinations, as fabricated, sorted by total equivalent area. Crosses indicate the selected combination, the right column indicates the total **FeCap** area obtained by summing the areas of selected capacitors. Duplicate combinations leading to the same total area are omitted. See also [Table C.2](#) for the 2T5C variant.

300 71×10^{-15}	400 126×10^{-15}	400 126×10^{-15}	400 126×10^{-15}	550 238×10^{-15}	←Capacitor Diameter (nm) ←↓Equivalent area (m^2)
					0
X					71×10^{-15}
			X		126×10^{-15}
X			X		196×10^{-15}
		X	X	X	238×10^{-15}
X				X	251×10^{-15}
X		X	X		308×10^{-15}
			X	X	322×10^{-15}
	X	X	X	X	363×10^{-15}
X			X	X	377×10^{-15}
X	X	X	X		434×10^{-15}
		X	X	X	448×10^{-15}
X		X	X	X	489×10^{-15}
	X	X	X	X	560×10^{-15}
X	X	X	X	X	615×10^{-15}
					685×10^{-15}

TABLE C.2: Possible 2T5C combinations, as fabricated, with the same methodology as [Table C.1](#) for the 2T4C variant.