



HAL
open science

Non Volatile Operators Emulation Platform

Alban Nicolas, Cédric Marchand, David Navarro

► **To cite this version:**

Alban Nicolas, Cédric Marchand, David Navarro. Non Volatile Operators Emulation Platform. NANOARCH '23: 18th ACM International Symposium on Nanoscale Architectures, Dec 2023, Dresden Germany, France. pp.1-6, <10.1145/3611315.3633252>. <hal-04984635>

HAL Id: hal-04984635

<https://hal.science/hal-04984635v1>

Submitted on 10 Mar 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Non Volatile Operators Emulation Platform

Nicolas Alban*, Marchand Cédric*, Navarro David*

*Univ Lyon, Ecole Centrale de Lyon, CNRS,

INSA Lyon, Université Claude Bernard Lyon 1,

CPE Lyon,

INL, UMR5270, 69130 Ecully, France

alban.nicolas@ec-lyon.fr

Abstract—The Von-Neumann bottleneck is one of the biggest problem to achieve higher computing performances and energy efficiency, especially in data centric applications. One of these application, the Internet of Things (IoT), is expanding at an impressive pace and is relying massively on sensors with limited energy to work. The emergence of new CMOS compatible technologies like ferroelectric field effect transistor (FeFET) allows to design new kind of Logic-in-Memory (LiM) operators. These non-volatile operators are expected to help overcoming the Von-Neumann Bottleneck as it will retain information thanks to the non volatility of the FeFET. The amount of data transfer will decrease thanks to this particularity. However, design of new computing operators is a long process. It is important to know if these operators are interesting to pursue into further development, which is why they need to be evaluated. To this extent, we present in this article a Non-volatile Operators Emulation Platform. the platform is based on the COMET RISC-V processor and use emulation to evaluate the impact of new non-volatile memories based operator on the processor performances in terms of number of instruction.

Index Terms—FPGA, RISC-V, FeFET, Emulation, Logic in Memory.

I. INTRODUCTION

For a few years now the use of Internet of Things (IoT) devices as well as Artificial Intelligence (AI) have been in the center of attention regarding electronic. This lead to an always growing quantity of data that need to be treated and secured. It is forecasted that the number of active connected IoT objects will achieve 30.9 Billion in 2025 [1]. The recent progress in processor technology improving the speed as well as power consumption has revealed a problem in the Von-Neumann and Harvard architectures. The separation of the computation unit and the memory has become the limiting factor in term of performances [2]. Also, memory transfers are representing one of the biggest part in the power consumption of the processor. Memory transfer alone would take more than half of the power budget of the processor [3]. This problem is now more important since the focus of new algorithms are mainly data intensive application or IoT devices, where the power supply is really small.

To overcome these challenges, numerous solutions are explored. One of them is the Logic In Memory (LiM) paradigm that appeared in the 70s [4]. The paradigm gained a lot of popularity in the recent years thanks to non-volatile emerging technologies that allow a tight integration with CMOS technologies such as ferroelectric field effect transistor (FeFET) [5] or spin-transfer torque magnetic tunnel junctions (STT-MTJs) [6]. LiM could be applicable to both AI and IoT, enabling us to mitigate the effects of the Von-Neumann bottleneck. However, there is a strong need of testing the new LiM devices that will be designed based on non-volatile technology.

In this article, we propose a RISC-V based platform that is dedicated to the evaluation of non-volatile operator designed around non-volatile emerging technologies and made for LiM. In our platform, these technologies are emulated with a behavioral model which is included in the form of new instruction inside the execution flow of the RISC-V processor. The Emulation is a method based on mimicking one technology or electronic component inside an other electronic component. This method allow us to be faster and cheaper than circuit design. It will also induce less issues than simulating emerging technologies with model on Cadence for a whole system like a processor. In the literature, the Emulation is mainly used regarding actual memory, like cache memory and main memory [7]–[9]. This article proposes a different approach that centers around the way in which the emulation of new LiM devices is done thanks to the help of the flexibility of RISC-V. Also, the non-volatility is used as a tool to store specific constants and allow reprogrammability of our devices. This is why it is not really comparable to these others article more in the way of emulating a partial, or complete memory, that would be non-volatile. Due to the high complexity of emulating the FeFET based component in a precise way [10], the choice has been made to use a model card described in sectionIII as a base for emulation. The platform is based on COMET [11] a RISC-V processor based on C++ High Level Synthesis (HLS) with an explicit pipeline.

We propose to evaluate the benefit of the integration of FeFET based operator inside the data-path of the RISC-V processor. The results shown in this paper are from the simulated version of our platform and are based on 28nm FeFET [5] based logic gates [12]. The design of theses operators would allow the addition of a security layer inside IoT nodes for

algorithm with publicly known constant since storing them directly will not impact the safety of our system. Evaluations are done using the Advanced Encryption Standard (AES) cryptographic algorithm and in particular the new non-volatile operators are placed in the Galois Field multiplication used in this algorithm. This Paper shows that the Emulation platform is an effective tool to evaluate newly designed operators. It shows an improvement by a factor of 4X in term of execution speed compared to the same program without any dedicated instruction. It also shows that the non-volatile version of the operator doesn't degrade the performance of the processor compared to the volatile version of the operator.

The rest of this article is organized as follow : Section II, you will find background related regarding FeFET, RISC-V instruction set architecture as well as hardware emulation. Section III presents the emulation platform. Section IV details the way you can use RISC-V to emulate new FeFET based operators with the implementation of the Galois field multiplication used in the AES cryptographic algorithm as a case study. Finally, Section V concludes the article.

II. BACKGROUND AND RELATED WORKS

A. RISC-V

RISC-V is an open instruction set architecture (ISA) created in 2015 in Berkeley ref création de RISC-V. It is designed to be flexible with its construction. It has a base instruction set on 32bits of data called "RV32I" on which it is possible to add some extension like the "M" extension that is focused around multiplication and division. The ISA is made around some template of instruction as shown in Fig.1 that give the definitions of every type of instructions bit by bit for the basic instruction set architecture. The R-type instruction where the R stands for Register is the main type regarding instruction doing computation using registers with two registers being used for the operands and a third one used to store the result. This I-type instruction is for instructions with immediate and load instruction, S-type is for store instruction, B-type is for branching, the U format is for instructions using upper immediate which are 20 bits integers, finally the J-type is for the Jump instructions.

To describe the parts visible in the Fig.1 in details: "rs" parts are for source register, "rd" for the destination register, "imm" stands for an immediate data to be used in the operation, the opcode gives information about what kind of instruction is going to be used whether it is loading data, computation with an immediate or between two registers from the CPU, and "funct" stands for which function is going to be used, allowing to make to a maximum of 1024 different functions per opcodes. The presence of Opcodes also let some free space to design custom instructions with dedicated Opcodes like 0x0B.

The RISC-V ISA is interesting because it is free and opened. Thanks to that, the community is really active and is developing numerous extensions. One of them regarding cybersecurity is on the way to get standardized at the moment [14], this one example is just a part of how users of the ISA

are willing to develop it freely. This extension shows that the tools and field we are trying to develop are relevant to the community.

B. FeFET based logic in memory

This article focuses on the utilization of FeFET based on 28nm SLP technology from GLOBALFOUNDRIES as the principal technology for our tests [5]. Non-volatility serves as a viable solution to address memory wall issues and provides excellent optimization for IoT circuits, which are typically low power. Modern technologies now allow for Non-volatile SRAM to achieve the same efficiency as classical SRAM [15]. Processors developed with non-volatile memory demonstrate significantly improved performance compared to classical systems, particularly in low-energy environments [16]. This is primarily because the need to reload the context is managed in a closer environment, and thanks to non-volatility, intermediate calculations are preserved, unlike classical volatile processors that often require restarting computations from the beginning. As a result, this leads to both time and energy savings for the processor.

The interesting part here about FeFET is that you can store one of the operand previously to the computation and then use the second operand passing through the gate. This enables the fact of having gates with only one FeFET like shown with the dynamic NAND from [12]. This also allows to explore paradigm like in memory computing or Logic in Memory [17]. Logic in Memory is a paradigm that exist since the 1970 [4] and that is all about making memory able of doing some computation directly in the memory cell instead of sending it to the computation unit. It is envisaged as an efficient solution to overcome the Memory wall [18] as well as to reduce the power consumption of circuits [19].

C. Emulation

The evaluation of the newly designed non-volatile operators in this paper is conducted through the use of emulation. Emulation involves extracting information regarding technology parameters such as delay and power consumption and utilizing these metrics to replicate the behavior within a larger system. When compared to simulation, emulation proves to be an interesting alternative. While simulation provides precise results, the risk of encountering non convergence becomes more significant when simulating an entire processor with the integration of emerging technologies, especially as the number of FeFET increases. Also, according to [20], classical simulators like Gem5 would require 8 hours to simulate just one second of processing for an entire core operating at 200 Thousand Instructions per seconds. On the other hand, the use of ASICs (Application-Specific Integrated Circuits) allows more accurate results. However, the time and financial resources required for designing such circuits are often less flexible than desired.

The emulation makes it possible to perform a fast evaluation during the early stages of development regarding emerging technology based operators. Indeed, it remains cheaper and

31	27	26	25	24	20	19	15	14	12	11	7	6	0	
funct7				rs2		rs1		funct3		rd		opcode		R-type
imm[11:0]				rs1		funct3		rd		opcode		I-type		
imm[11:5]			rs2		rs1		funct3		imm[4:0]		opcode		S-type	
imm[12 10:5]			rs2		rs1		funct3		imm[4:1 11]		opcode		B-type	
imm[31:12]								rd		opcode		U-type		
imm[20 10:1 11 19:12]								rd		opcode		J-type		

Fig. 1: The different type of RISC-V instruction from [13]

less time consuming than manufacturing demonstrators to ensure proper functionality. In addition, design using emerging technologies is usually based on model that may cause simulation issues in design tool such as Cadence, which makes the simulation of scaled operation very long and difficult.

In the attempt to integrate novel computing operators designed with emerging non-volatile technologies, the initial idea was to approximate the hysteresis curves from the actual FeFET model [10] to obtain the most precise information. However, these models proved to be excessively complex and resource-intensive using a huge number of lookup tables (LUTs) and Digital signal processors (DSPs), even on the large FPGA that is going to be used in the future [21]. Therefore, in order to maintain flexibility and efficient resource usage on the Field Programmable Gate Array (FPGA) board, a simpler model depicting the behavior with delays and energy consumption are used for emulating the designed operators.

III. THE PROPOSED PLATFORM

The objective of the RISC-V based platform presented in this section is to evaluate new computing operators designed using emerging technologies in their early stage of development. This evaluation platform needs to be as generic as possible to allow designers to add or change both the operators and technology. The evaluated operator is integrated through emulation of its behavior in terms of timing for now and in terms of consumption for each type of operation it can perform. The overall evaluation platform is depicted in Fig. 2.

As presented in Fig. 2, the emulation platform is divided into two main blocks: the core manager and the RISC-V processor. These two main blocks interact with each other through a control bus and the Core Manager (CM) Access, that allows the CPU to tell the Core Manager when to go on a break and send data back to the computer. The whole platform can be controlled with a serial communication. The UART will also send the instruction and data to the memory part, which is a Dual port Ram (DPRAM).

A. Core Manager

The Core Manager uses four 8 bits commands to fully control the platform :

- Get the whole program from the computer through the UART to the DPRAM,
- Send part or all of the memory back to the computer,

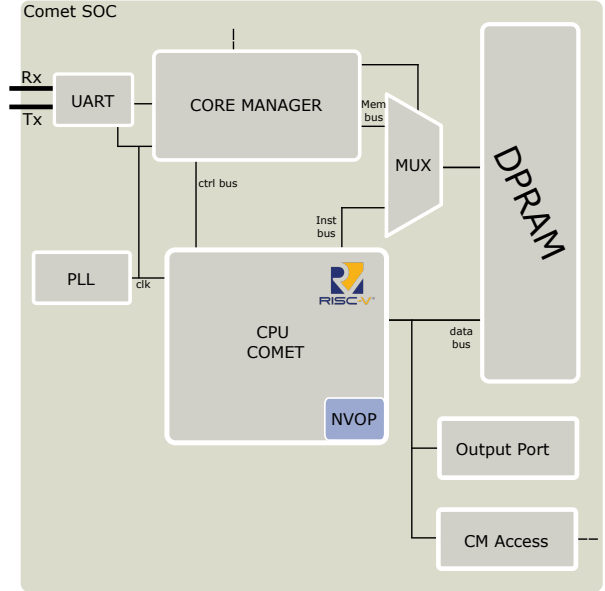


Fig. 2: Plan of the Non-Volatile Emulation Platform

- Start the CPU by sending an enabling signal through the control bus,
- Put the CPU on halt if a signal is received from the CM Access part.

The platform is fully managed using these four commands and ensure a proper use of the RISC-V processor. In addition, the ability to send a pause command at any moment makes it possible to get information after critical computation.

B. A RISC-V processor : COMET

The main computation unit in the platform is the COMET RISC-V processor [11]. The COMET processor is described in C++ language and is able to be synthesized for FPGA using High Level Synthesis (HLS) tools, allowing us to get either a VHDL or a Verilog description of the processor. The C++ definition of the processor makes it easy to modify as well as really understandable with an explicit description of the five stage pipeline.

The processor is a five stages pipeline. It is based on the RV32-I Instruction set architecture which is the smallest among the one proposed by RISC-V. In addition, the cache memory has been removed from the CPU to stay consistent with area and energy constraint devices as in the IoT context.

C. Non-volatile operators

The way the non-volatile operators are added into the RISC-V is by creating new instructions for the ISA. Each instruction being dedicated to the use of one operator. The newly made instructions are then placed at the point in the C code where the operator is going to be used thanks to inline assembly simplifying the process to test operators since the need to modify the compiler is removed. The first step to make the designed non-volatile cells into instructions was to adapt the delays of the model for the cell used to the way the processor is working. In the case of the proposed platform, the COMET processor is working at 100MHz, meaning the operators need to be used according to a $10ns$ schedule. If the time needed to process the data with the new operator is greater than $10ns$ the number of processor cycles needs to increase accordingly. For example, if the operator takes $23ns$ to process the data then a total of three core cycles will be needed. Also, to not disturb the pipeline of the processor, during the moments the operators are used, the processor will be put into a stall state for the right amount of cycles to mimic the time needed by the designed cells. This allows the gathering of information for the efficiency regarding computational speed to be collected. For the part regarding power consumption, the platform possesses counters monitoring the number of transition for each bit of non-volatile data. This is also done for the reading part of the FeFET. The values of these counters are then sent to the DPRAM at dedicated addresses and then sent to the computer for analysis to recover the energy consumed by the operator during a particular program.

IV. EXPERIMENTS AND RESULTS

A. Presentation of the Experimental setup

The decision was made to implement a segment of a cryptographic algorithm as a use case for multiple reasons. Firstly, most of cryptographic algorithm use publicly available constants. Secondly, these constants remain the same over a long period of time. We are still using cryptographic algorithms such as the AES, which has been standardized in 2001. This makes the constants good targets to be stored inside the newly designed operators. Finally, most of the cryptographic algorithms are using special arithmetic that generic processor does not possess specific instruction for, since cryptographic computation defer from classical operation. In this article, the newly designed operator is developed for the multiplication into the Galois field used in the AES. In such operation, the result must stay in the field. To ensure this property, an irreducible polynomial of the field is used to reduce any result that is out of the field. The operator is based on already existing circuit [22] dedicated to efficiently do it, where it has been modified to include some FeFET. The implementation of multi-bit multiplication uses a serial circuit, meaning that 8 cycles are needed to perform an 8-bit Galois field multiplication. Developing a non-volatile operator for such computation will reduce amount of core cycles needed to process the data, making the modified processor faster

Concerned Part	Actual Values (in term of core cycles)	Prospected Values
Multiplication Circuit	8	1
Reading into FeFET	26	1
Writing into FeFET	52	10

TABLE I: Information about the used parameters

compared to basic version. In addition, when the standard will change, the processor instruction can be reconfigured to be adapted to the new algorithm, increasing the lifetime of the system.

To implement this operation, two new instructions are added to instruction set architecture from RISC-V:

- LoadPoly : An instruction setting up the nine bits of the irreducible polynomial in our platform, where there is a dedicated register for it inside our processor.
- GaloisMul : This instruction realizes the Galois Field multiplication of two eight bits integers. It is important to know how many bits are used, since the number of cycle needed to complete the multiplication is related to the length of the data.

Each new instruction exists in two version, one for the volatile operation and one for the non-volatile operation based on FeFET. For the LoadPoly instruction, the change is done regarding the monitoring of the changes on all the FeFET, whether each bit transitions. For the GaloisMul instruction, the difference relies on the number of cycle the COMET processor will be in its stall state. Here, there is the need to adapt the stall time to the integrated operator. For example, the processor needs to be placed in its stall state for 8 cycles for an 8-bit Galois field multiplication. It is Also important to take into account the time needed to write values inside FeFET or read for FeFET when it is needed.

The experiments will take place with two different set of values. The first one is based on the circuit design presented before in this section, where the FeFET data used are based on the Cadence simulation results based on $500nm$ square FeFET from [12]. These results are way less optimal compared to what is forecasted to be in the future with the reading into a FeFET taking $5ns$ and the writing a $100ns$ [23], which translate into a single core cycle to read and ten to write. The circuit would also be optimized, allowing the data to be processed in parallel which would change the eight cycles needed to a single one. All these parameters are summarized in Table I where the unit displayed is the core cycle, which is equal to $10ns$.

Finally, the operator will be used in the decryption part. Indeed, for the encryption the value with which the operands need to be multiplied are integers between 1 and 3 making the Galois Field multiplication irrelevant compared to a left shift that is already defined in the RISC-V ISA.

B. Results and discussion

After conducting the tests with the presented combinations of parameters, the results shown in Figure 3 illustrate the

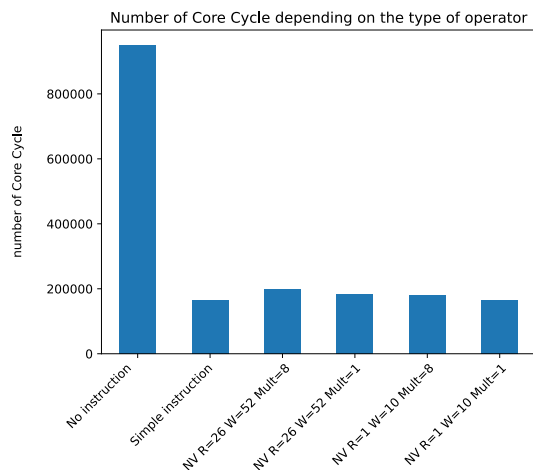


Fig. 3: Graph of the Number of core cycle for one decryption for Different Types of Parameters Used in the Non-Volatile Operator

significant simplification of the processor’s workload by incorporating a new instruction. The following can be read, the amount of core cycles required for a single deciphering operation is reduced by a factor of five when comparing the “No Instruction” test where no new instruction were added with the “simple instruction” where the Galois Field Multiplication is added without any non-volatile component. The difference between the volatile and non-volatile versions of our instructions depends on the optimization level of the operator. Currently, with the reading of a FeFET operation taking 26 cycles and a multiplication taking 8 cycles, represented in Fig3 by “NV R=26 W=52 Mult=8”, there is an overhead of approximately 20% compared to the volatile version of the instruction. However, as depicted in the graphs in Figure 4, this overhead diminishes over time. In fact, with the most optimized technology projections, the one where the operator is fully parallel and with the prospected value of the FeFET shown in Table I, the non-volatile version of the operator outperforms the simple instruction. This improvement becomes evident after using the non-volatile version of the deciphering operation three times, as shown in Table II.

This improvement can be attributed to the elimination of the need to reload the polynomial, resulting in faster and more efficient computations. The relatively small amelioration is due to the fact that the amount of data saved represents only a nine-bit polynomial, which is a relatively small amount of data.

V. CONCLUSION

In this paper, we have presented a RISC-V based emulation platform able to quickly tests the efficiency of LiM devices based on non-volatile memory by introducing dedicated instructions, that emulate the newly designed operators using a model, to the basic RISC-V ISA. Also, the results of our Galois Field Multiplication instructions reveal a significant re-

duction in the number of core cycles required for the processor compared to a version without any dedicated instructions. This demonstrates the efficiency gained by incorporating specialized instructions for Galois Field Multiplication. By comparing the two versions of the instruction—one non-volatile and the other volatile—we observed that the non-volatile version introduces only a small overhead using the current technique. This indicates that the non-volatile nature of the operator does not significantly impact performance. However, we took it a step further and optimized the non-volatile version, leading to a notable reduction in the overhead. In fact, the optimized non-volatile version became even more efficient than the volatile counterpart, showcasing the advantages of incorporating non-volatile operators.

Our future research steps involve transitioning from simulation to implementing the platform directly with an FPGA. This will make it possible to use the same platform to develop and test a first ASIC demonstrator of non-volatile operators in order to extract the real values of latency and energy consumption from manufactured operator only. Additionally, we plan to expand our investigation by exploring the integration of more non-volatile parts with the addition of the evaluation of the Substitution box, which is also an important part of the AES algorithm. This will provide a comprehensive understanding of the impact and benefits of non-volatile operators in cryptographic algorithms.

ACKNOWLEDGMENT

This work was carried out in the framework of the SECRET project supported by the French “Agence Nationale de la Recherche” under project number ANR-20-CE39-0006.

REFERENCES

- [1] “Global IoT and non-IoT connections 2010-2025.” <https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/>.
- [2] D. Efnusheva, A. Cholakoska, and A. Tentov, “A SURVEY OF DIFFERENT APPROACHES FOR OVERCOMING THE PROCESSOR-MEMORY BOTTLENECK,” *International Journal of Information Technology and Computer Science*, p. 151, Apr. 2017.
- [3] S. Borkar and A. A. Chien, “The future of microprocessors,” *Commun. ACM*, vol. 54, pp. 67–77, May 2011.
- [4] H. S. Stone, “A Logic-in-Memory Computer,” *IEEE Trans. Comput.*, vol. 19, pp. 73–78, Jan. 1970.
- [5] S. Beyer, S. Dünkel, M. Trentzsch, J. Müller, A. Hellmich, D. Utess, J. Paul, D. Kleimaier, J. Pellerin, S. Muller, J. Ocker, A. Benoist, H. Zhou, M. Mennenga, M. Schuster, F. Tassan, M. Noack, A. Pourkaramati, F. Müller, and S. Slesazek, “FeFET: A versatile CMOS compatible device with game-changing potential,” pp. 1–4, May 2020.
- [6] E. Garzón, B. Zambrano, T. Moposita, R. Taco, L.-M. Pröcel, and L. Trojman, “Reconfigurable cmos/stt-mtj non-volatile circuit for logic-in-memory applications,” in *2020 IEEE 11th Latin American Symposium on Circuits Systems (LASCAS)*, pp. 1–4, 2020.
- [7] G. Zhu, K. Lu, X. Wang, X. Zhou, and Z. Shi, “Building Emulation Framework for Non-Volatile Memory,” *IEEE Access*, vol. 5, pp. 21574–21584, 2017. Conference Name: IEEE Access.
- [8] F. Wen, M. Qin, P. V. Gratz, and A. L. N. Reddy, “FPGA-based Hybrid Memory Emulation System,” *arXiv:2011.04567 [cs]*, Nov. 2020. arXiv: 2011.04567.
- [9] S. Ruffini, L. Caronti, K. S. Yıldırım, and D. Brunelli, “NORM: An FPGA-based Non-volatile Memory Emulation Framework for Intermittent Computing,” *J. Emerg. Technol. Comput. Syst.*, Feb. 2022. Just Accepted.

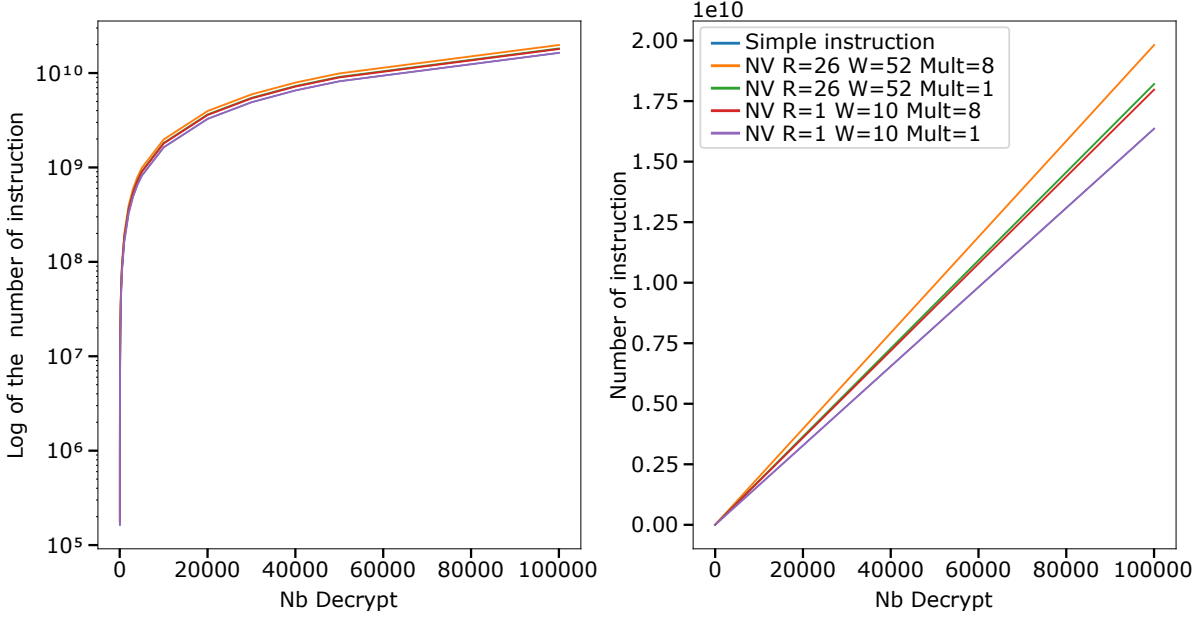


Fig. 4: Graph representing the number of core cycles needed for the deciphering operation depending on the number of decryption. (a) show the results in a logarithmic scale and (b) shows the raw results

Nb Decrypt	Simple Instruction (raw value)	NV R=26 W=52 Mult=8	NV R=26 W=52 Mult=1	NV R=1 W=10 Mult=8	NV R=1 W=10 Mult=1
1	163967	34611	18483	16137	9
3	491197	103717	55333	48379	-5
5	818427	172823	92183	80621	-19
10	1636502	345588	184308	161226	-54
30	4908802	1036648	552808	483646	-194
50	8181102	1727708	921308	806066	-334
100	16361852	3455358	1842558	1612116	-684
300	49084852	10365958	5527558	4836316	-2084
500	81807852	17276558	9212558	8060516	-3484
1000	163615352	34553058	18425058	16121016	-6984

TABLE II: Difference of cycle taken between the Simple instruction and the different version of the non volatile operator

- [10] M. Paul-Antoine, M. Cedric, N. David, and I. O'Connor, "Hardware Emulation of FeFET On FPGA," in *2022 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 380–385, July 2022. ISSN: 2159-3477.
- [11] S. Rokicki, D. Pala, J. Paturel, and O. Sentieys, "What You Simulate Is What You Synthesize: Designing a Processor Core from C++ Specifications," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, (Westminster, CO, USA), pp. 1–8, IEEE, Nov. 2019.
- [12] C. Marchand, I. O'Connor, M. Cantan, E. T. Breyer, S. Slesazek, and T. Mikolajick, "FeFET based Logic-in-Memory: an overview," in *2021 16th International Conference on Design Technology of Integrated Systems in Nanoscale Era (DTIS)*, pp. 1–6, June 2021.
- [13] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanovi, "The risc-v instruction set manual. volume 1: User-level isa, version 2.0," tech. rep., California Univ Berkeley Dept of Electrical Engineering and Computer Sciences, 2014.
- [14] "RISC-V cryptographic extension - home - RISC-V International." <https://wiki.riscv.org/display/HOME/Cryptographic+Extensions+TG>.
- [15] W.-X. You, P. Su, and C. Hu, "A New 8T Hybrid Nonvolatile SRAM With Ferroelectric FET," *IEEE Journal of the Electron Devices Society*, vol. 8, pp. 171–175, 2020. Conference Name: IEEE Journal of the Electron Devices Society.
- [16] Y. Wang, Y. Liu, S. Li, D. Zhang, B. Zhao, M.-F. Chiang, Y. Yan, B. Sai, and H. Yang, "A 3us wake-up time nonvolatile processor based on ferroelectric flip-flops," pp. 149–152, Sept. 2012.
- [17] I. O'Connor, M. Cantan, C. Marchand, B. Vilquin, S. Slesazek, E. T. Breyer, H. Mulaosmanovic, T. Mikolajick, B. Giraud, J.-P. Noël, A. Ionescu, and I. Stolichnov, "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)*, pp. 180–183, Oct. 2018. ISSN: 2324-8440.
- [18] X. Zou, S. Xu, X. Chen, L. Yan, and Y. Han, "Breaking the von Neumann bottleneck: architecture-level processing-in-memory technology," *Sci. China Inf. Sci.*, vol. 64, p. 160404, Apr. 2021.
- [19] X. Yin, A. Aziz, J. Nahas, S. Datta, S. Gupta, M. Niemier, and X. S. Hu, "Exploiting ferroelectric FETs for low-power non-volatile logic-in-memory circuits," in *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, Nov. 2016. ISSN: 1558-2434.
- [20] SanchezDaniel and KozyrakisChristos, "ZSim," *ACM SIGARCH Computer Architecture News*, June 2013. Publisher: ACM PUB27 New York, NY, USA.
- [21] "Netfpga-sume fpga development board." <https://www.xilinx.com/products/boards-and-kits/1-6ogkf5.html>.
- [22] P. Scott, S. Tavares, and L. Peppard, "A fast vlsi multiplier for gf(2^m)," *IEEE Journal on selected Areas in Communications*, vol. 4, no. 1, pp. 62–66, 1986.
- [23] S. Dutta, H. Ye, W. Chakraborty, Y.-C. Luo, M. S. Jose, B. Grisafe,

A. Khanna, I. Lightcap, S. Shinde, S. Yu, and S. Datta, "Monolithic 3d integration of high endurance multi-bit ferroelectric fet for accelerating compute-in-memory;" in *2020 IEEE International Electron Devices Meeting (IEDM)*, pp. 36.4.1–36.4.4, 2020.