



HAL
open science

FeFET based Logic-in-Memory: an overview

Cédric Marchand, Ian O'Connor, Mayeul Cantan, Evelyn T Breyer, Stefan Slesazeck, Thomas Mikolajick

► **To cite this version:**

Cédric Marchand, Ian O'Connor, Mayeul Cantan, Evelyn T Breyer, Stefan Slesazeck, et al.. FeFET based Logic-in-Memory: an overview. 2021 16th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), Jun 2021, Montpellier, France. pp.1-6, 10.1109/DTIS53253.2021.9505078 . hal-03323883

HAL Id: hal-03323883

<https://hal.science/hal-03323883v1>

Submitted on 23 Aug 2021

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.

FeFET based Logic-in-Memory: an overview

Cédric Marchand*, Ian O'Connor*, Mayeul Cantan* Evelyn T. Breyer†, Stefan Slesazek†, Thomas Mikolajick†‡

* Univ Lyon, Ecole Centrale de Lyon, CNRS,
INSA Lyon, Université Claude Bernard Lyon 1,
CPE Lyon,
INL, UMR5270, 69130 Ecully, France
cedric.marchand@ec-lyon.fr

†NaMLab gGmbH
Dresden, Germany
‡Chair of Nanoelectronics, IHM,
TU Dresden, Germany
stefan.slesazek@namlab.com

Abstract—Emerging non-volatile memories are getting new interest in the system design community. They are used to design logic-in-memory circuits and propose alternatives to von-Neuman architectures. Hafnium oxide-based ferroelectric memory technology, which is fully compatible with CMOS technologies is particularly interesting for logic-in-memory designs. Indeed, this compatibility leads to various possibilities for fine-grain logic in memory applications where the memory capable element is tightly integrated with the transistors in the system. Non-volatile and energy efficient computing for Internet of things and embedded artificial intelligence are among the potential applications for this technology.

In this article, we focus on ferroelectric field-effect transistors (FeFET) and present an overview of three different fine-grain logic-in-memory possibilities with FeFETs: custom operation designs, reconfigurable circuits and a hybrid memory element accessible by content or by address. All presented circuits have been designed within a test chip using $28nm$ technology provided by GLOBALFOUNDRIES.

Index Terms—FeFET design, Logic-in-Memory, Look up table, Non-volatile logic gates, Ternary content address memory, TCAM, TC-MEM.

I. INTRODUCTION

Data processing and evaluation is a key aspect of today's digital systems with the explosions of Internet of Things IoT and Artificial Intelligence (AI) paradigm. Von-Neumann computing architectures have shown an important bottleneck that limits the data throughput and the power consumption reduction of processing systems: the separation between processing and storage of the architecture leads to data transfer congestion. Data transfer in von-Neumann architectures corresponds to the most power consuming part in actual computing systems. With the growth of concept such a IoT and AI, the amount of collected and analyzed data is becoming massive (several zettabyte per year) and it will continue to increase. Consequently, the classical computing paradigm is arriving to a limit and new direction have to be explored.

To overcome this bottleneck, Logic-in-Memory (LiM) concept is a serious candidate because it reduces the number of data transfers between the system memory and the computing core. The concept has been described in 1970 in [1] but is recently getting new interest thanks to the emergence of non-volatile memories achieving higher integration with CMOS transistors than old memory technologies [2], [3]. Another factor of the interest for LiM designs relies on

various applications, such as AI [4], cryptographic algorithm implementations [5] or near sensor computing for IoT [6], this concept can address. Resistive technologies (ReRAM) [3], Magnetic technologies (MRAM) [7], Phase change memories [8] or Ferroelectric technologies (FeRAM and FeFET) [9] are listed as technologies for LiM applications [10], [11].

LiM concept can be divided in different groups depending on the location of the computation but also on the architecture of the memory itself [9]. These different concepts are named and known as: Processing in memory, In memory Computing, coarse-grain LiM where the non-volatile memory is arranged to a matrix and placed sufficiently close to the processor to be used with the same latency than L1 cache, and fine-grain LiM where the non-volatile memory is tightly integrated with the logic design and can be placed inside computing operations.

In this article, we focus on hafnium oxide-based ferroelectric devices [12], which are a very promising for fine-grain LiM design with a high efficiency in terms of area and energy. This technology is fully compatible with CMOS manufacturing process and it is possible to tightly integrate the ferroelectric layer with the transistors to create Ferroelectric Field Effect Transistors (FeFETs). Whereas coarse-grain applications and examples has been already demonstrated using FeFETs [13], fine-grain LiM exploration receive less attention [14]. A deep exploration of fine-grain LiM possibilities using FeFETs shows three different uses:

- Custom operation with constant values stored inside FeFETs.
- Re-configurable logic [15].
- Innovative memory elements for Computing-in-Memory (CiM) and LiM.

This article proposes an overview of the three fine-grain LiM opportunities brought by FeFETs based designs and discuss possible applications. All presented circuits have been designed within a test chip using $28nm$ technology provided by GLOBALFOUNDRIES. The rest of this article is organized as follow: in Sec. II, we presents the technology, its behavior and explain how we use FeFET in designs. In Sec. III, the elementary logic gates used to create custom logical operations are presented. Sec. IV details the design of Look-Up-Table (LUT) based reconfigurable logic gate designed with FeFETs. In Sec. V, we present a hybrid memory element accessible

either by content or by address. Sec. VI discuss applications opportunities brought by the presented FeFET designs. Finally, Sec. VII conclude the paper and gives future directions.

II. TECHNOLOGY AND RELATED WORK

A. FeFET technology

FeFETs have already been presented in detail in [16]. Basically, it can be seen as a regular transistors with an additional ferroelectric capacitor connected to the gate as presented in Fig. 1.a. Depending on the polarization of the capacitor, the transistors will switch between its *OFF* and its *ON* state at a different gate voltage (V_g in Fig. 1.a). To change the polarization of the ferroelectric capacitor, a strong positive/negative voltage is applied to gate of the FeFET. Fig. 1.b presents the two characteristics of the FeFET transistor. The left one corresponding to low threshold volatge state (*LVT*) and the right one to the high threshold volatge state (*HVT*).

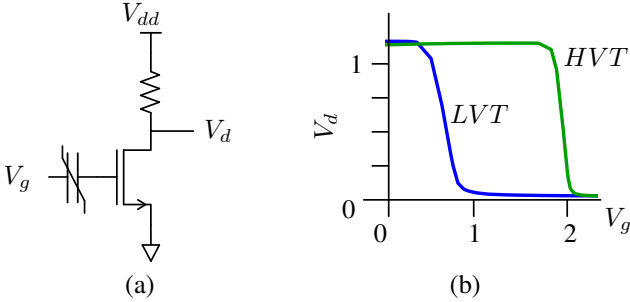


Figure 1: (a) Basic FeFET circuit with a passive load and (b) the voltage characteristic of the FeFET showing the different behavior depending on the polarization of the ferroelectric capacitor.

Thank to this characteristic, a single FeFET realizes a *NAND* operation between a logical value stored inside the FeFET and the evaluating input presented at V_g . Fig. 2 highlights this operation with a simulation of the schematic presented in 1.a.

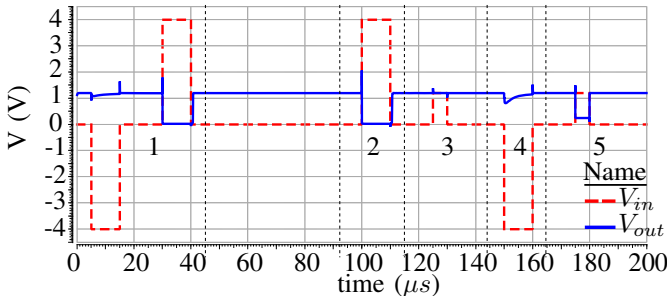


Figure 2: Simulation of the FeFET NAND gate.

5 different areas can be seen in Fig. 2:

- 1) Initialization of the FeFET model.
- 2) A $+4V$ write pulse on V_{in} to store logical 0.
- 3) A $1.2V$ read pulse on V_{in} to evaluate the operation.

- 4) A $-4V$ write pulse on V_{in} to store logical 1.
- 5) A $1.2V$ read pulse on V_{in} to evaluate the operation.

Comparing the area 3 and 5 enable us to see the *NAND* gate truth table.

In the rest of this paper, the following parameters are used for the FeFET size: Length (L) = $500nm$, Width (W) = $500nm$. This corresponds to large devices but for which the Preisach model has been fitted to ASIC demonstrator [12]. All results have been confirmed using ASIC chip measurements.

B. Design considerations

To design specific logical operation using FeFETs and its memory capacity, it is required to design elementary logic gates [12]. These basic logic gates are *NAND*, *OR*, *XOR* and *AND*. These elementary logic gates can be used to synthesize any complex operations. However, only *NMOS* type FeFET is currently available in manufacturing. This implies the design of dynamic logic gates as presented in Fig 3 to limit the power consumption.

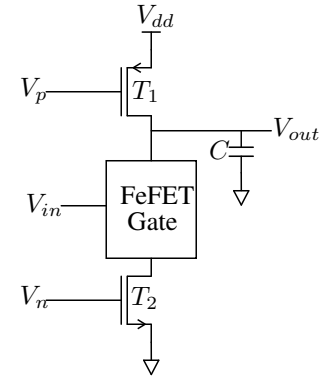


Figure 3: Generic schematic of dynamic logic gate using FeFET transistors.

In Fig. 3, the size of transistor T_1 is: $W = 130nm$, $L = 50nm$; and the size of transistor T_2 is: $W = 80nm$, $L = 36nm$. These dimensions are the smallest that respect design rules of the GLOBALFOUNDRIES $28nm$ technology. C represents the load capacitance of the gate. For simulation, we used $C = 1pF$, on ASIC the load capacitance corresponds to the sum of all parasitic capacitances between the output of the gate and the oscilloscope used for measurements. Use of the logic gates in this dynamic design follow 4 steps:

- 1) The constant operand is written in the FeFET.
- 2) The output (C) is charged to V_{dd} : T_1 opened, T_2 blocked
- 3) The logic gate is activated: T_1 blocked, T_2 opened
- 4) The logic gate is evaluated using the second operand

Thanks to these 4 steps, there is no direct link between the power and the ground. However, this requires 2 clock signals and the overall logic operation will take more time.

III. FINE-GRAIN LOGIC IN MEMORY DESIGN : CUSTOM OPERATIONS

The first possibility with FeFETs is to synthesize operations where constant operands are stored inside the ferroelectric

layer of the FeFET and the others are evaluating inputs. For this, elementary logic gates are necessary and some of them have already been presented [12], [9]. Fig. 4 shows the schematic of these elementary FeFET-based logic gates.

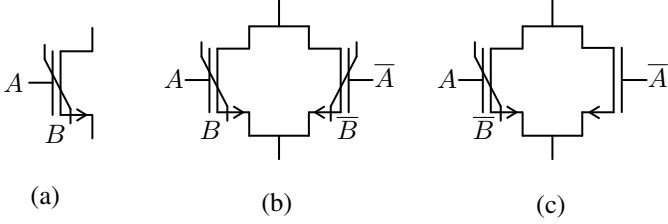


Figure 4: Schematic of the three elementary logic gates: (a) *NAND*, (b) *XOR* and (c) *AND*

The *NAND* gate, already presented in Sec. II can also implement *OR* operation by applying the De Morgan rule: $\overline{A \cdot B} = \overline{A} + \overline{B}$. The *XOR* gate is achieved by connecting two FeFET transistors in parallel [9] as shown in Fig. 4.b. Already presented as a ternary content addressable memory in [17], it is also possible to use this gate in logic synthesis. The simulation is presented in Fig. 5 and demonstrates the *XOR* operation.

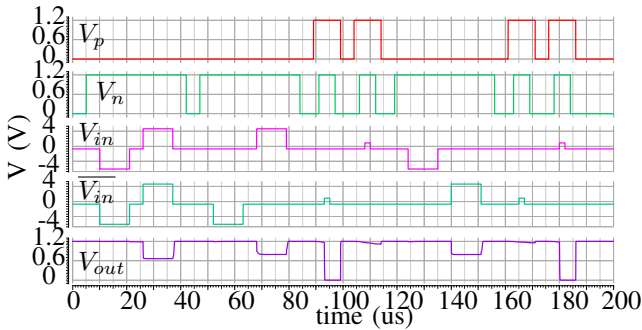


Figure 5: Simulation of the FeFET *XOR* gate.

The last elementary logic gate is the *AND* gate, the first option to design it is to apply a CMOS inverter at the output of the *NAND* gate. However, it is also possible to connect a *NMOS* transistor in parallel with a FeFET as presented in Fig. 4.c. Fig. 6 confirms the *AND* behavior.

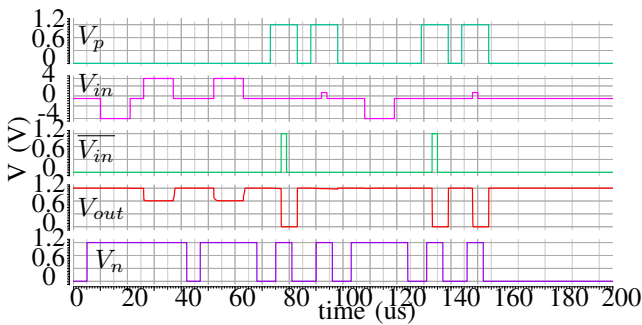


Figure 6: Simulation of the FeFET *AND* gate.

Using the presented logic gates, it is possible to synthesize any function with one non-volatile operand stored inside the

FeFETs. This enables us to avoid reloading this data each time the system is powered and ensures energy efficiency. The advantage of this concept is to reduce the number of data transfer from the main system memory in application such as IoT and AI.

IV. DESIGN GENERIC AND RE-CONFIGURABLE LOGIC GATE

The second concept corresponds to the design of generic structures that can be configured to perform different operations. The configuration will be stored inside the non-volatile memory. This approach is exactly the one used inside today's SRAM FPGAs. In [15], the proposed architecture used the FeFET only as a memory to replace SRAM cells and keep the classical architecture of look up table (LUT). The LUT structure stays classical and can be represented by the Figure 7.a.

However, by taking into account that the FeFET is actually also a logical element and not just a memory element, it is possible to design a LUT which is more efficient in terms of area and energy consumption as presented in [18] and in this work. As shown in Fig. 7.b, the memory capable FeFET and the first multiplexer stage of the LUT are merged into a single layer.

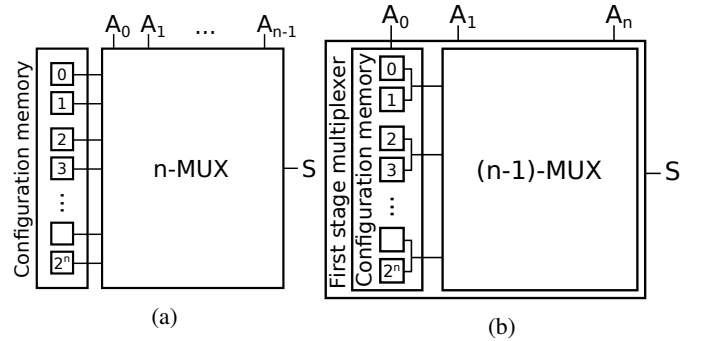


Figure 7: Architecture of a classical Look up Table (a) and architecture of a LUT using FeFET as memory and first stage multiplexer (b)

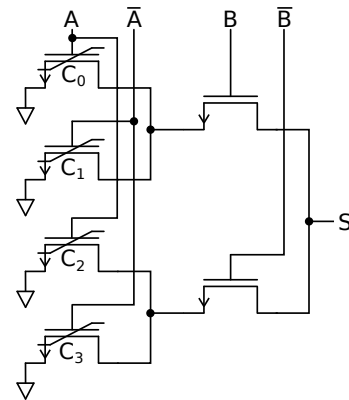


Figure 8: 2-input LUT using FeFET in the architecture proposed in the Figure 7.b.

All possible configurations of this 2-inputs LUT presented in Fig. 8 have been tested successfully and its global behavior can be described by the Equation 1 :

$$\bar{S} = C_0.A.B + C_1.\bar{A}.B + C_2.A.\bar{B} + C_3.\bar{A}.\bar{B} \quad (1)$$

From the Equation 1, all logical operations with 2-input are possible. In addition, it is easy to design a N -inputs LUT using FeFET transistors as memory and logical element. It consists in doubling the $N - 1$ inputs LUT and linked them using a multiplexer for the last input. Considering this, a N -inputs LUT will contain 2^N FeFET as configuration memory and first stage multiplexer (for the first input), followed by a classical $(N - 1)$ multiplexer (for the other inputs).

Based on the transistor count presented in [15], the LUT structure proposed in the Figure 7b reduces the number of transistor by 2^N transistors for a N -input LUT. Thus, we achieve a basic transistor count of 2^{N+1} transistors instead of 2^{N+2} which is a 50% area reduction (in term of number of transistors). In addition, only half of the FeFET transistors will be evaluated when the LUT is used which will also reduce the power consumption of the LUT compared to the classical LUT architecture.

V. HYBRID MEMORY ELEMENT

Emerging technologies such as resistive RAM (ReRAM) or Ferroelectric technologies (FeRAM and FeFET) have been used to design compact ternary content addressable memories (TCAM) [19], [17]. However, these circuit can only be used as TCAM and it will be interesting for some application to combine memory functionalities (i.e. TCAM and RAM). Such combination as been proposed using CMOS technology in [20]. Here, we propose a new hybrid memory element, based on FeFETs which combine TCAM and RAM functionalities: the TC-MEM.

A. The TC-MEM circuit

To design the 1-bit TC-MEM cell, two modes of operations are needed. The first mode of operation for the TC-MEM cell corresponds to normal memory mode. This is achieved with a single FeFET transistor as presented in Fig. 4.a. The second mode of operation corresponds to the TCAM mode, for which the structure is a *Xor* operation, is presented in Fig. 4.b.

It is possible to combine these two modes of operations by adding a NMOS transistor between the two FeFETs connected in parallel as presented in Fig. 9.

If the Mode input (M) is low ($M = 0$), the NMOS transistor is not conducting. Thus, only *FeFET2* can discharge the output line named *ML/WL* for match/word line (TCAM/memory mode): it is the memory mode. At the contrary, if M is high ($M = 1$), the NMOS is conducting. *FeFET1* and *FeFET2* are connected: it corresponds to the TCAM mode. The V_p input is used to precharge *ML/WL* line.

Fig. 10 shows the cadence simulation of the TC-MEM circuit. ASIC measurements confirmed this behavior and prove the dual functionality.

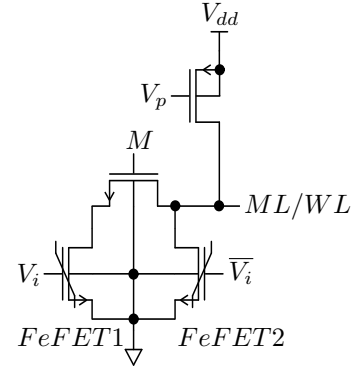


Figure 9: Schematic of the 1-bit TC-MEM cell.

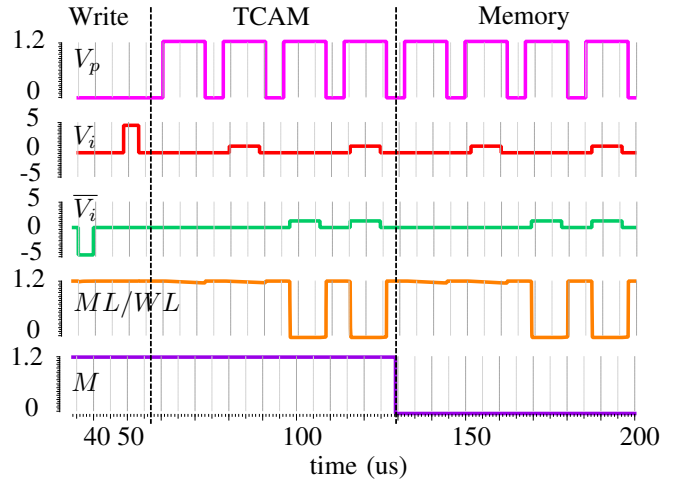


Figure 10: Simulation of the 1-bit TC-MEM cell with 0 stored in the cell.

In Fig. 10, a logical '0' is stored inside the cell. In TCAM mode, the match line (*ML*) is discharged when $V_i = 0$ and $\bar{V}_i = 1$ (i.e. the correct input is presented). When $V_i = 1$ and $\bar{V}_i = 0$, *ML* remains high.

In memory mode (i.e. $M = 0$), the word line (*WL*) gives the stored value if $\bar{V}_i = 1$. Otherwise, *WL* is high. Tab. I summarizes the complete behavior of the 1-bit TC-MEM circuit.

Table I: Summary of the 1-bit TC-MEM cell behavior with 1 stored inside

Mode	\bar{V}_i	V_i	<i>ML/WL</i>	Result
TCAM	0	0	1	Blocking don't care
	0	1	1	Don't match
	1	0	0	Match
	1	1	0	Passing don't care
RAM	0	0	1	-
	0	1	1	-
	1	0	0	Read
	1	1	0	Read

To scale the 1-bit TC-MEM to multi-bit cell, it is possible to connect multiple TC-MEM cells in series. In Fig. 11, the two 1-bit TC-MEM are serially connected with the line named *sc*.

An additional *NMOS* transistor (T_m) is added to the circuit to avoid the creation of a floating node between the lower and the upper bit cells in memory mode. In addition, T_m establishes the necessary ground connection to discharge the upper word line (WL_1) in memory mode (i.e., it is possible to read the upper bit A_1).

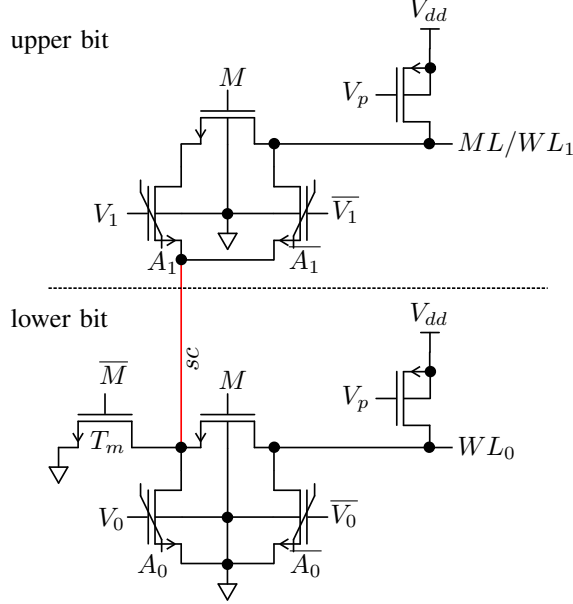


Figure 11: Schematic of the 2-bit TC-MEM circuit.

In TCAM mode ($M = 1$), the serial connection ensure that the match line (ML) output is high if one of the two 1-bit TC-MEM cell is opened (i.e. not passing). Considering that two FeFET connected in parallel realize a *Xor*, it is possible to summarize 2-bit TC-MEM circuit by Eq. 2:

$$ML = (V_1 \oplus A_1) + (V_0 \oplus A_0) \quad (2)$$

Eq. 2 implies that the match line will be low only if the correct value is presented for the entire 2-bit word.

The TC-MEM circuit can be generalized to an n -bit cell using serial connections. The number of transistors ($\#T$) needed to realize an n -bit TC-MEM cell is:

$$\begin{aligned} \#T &= n(2 \text{ FeFET} + 1 \text{ NMOS}) + (n - 1)(1 \text{ NMOS}) \\ &= 4n - 1 \end{aligned} \quad (3)$$

Eq. 3 does not take into account precharge transistors because their number will depend on the implementation of the full TC-MEM array. However, if only one output line is set for each bit of a word in a full TC-MEM array (as presented in Fig. 12), n precharge transistors will be needed.

VI. DISCUSSION ON APPLICATION OPPORTUNITIES

All the FeFET based circuits presented in this article can be used for diverse applications in constraint context such as the IoT for example. In addition, AI applications [21] and in memory computing [22] can also be addressed by FeFET based circuits.

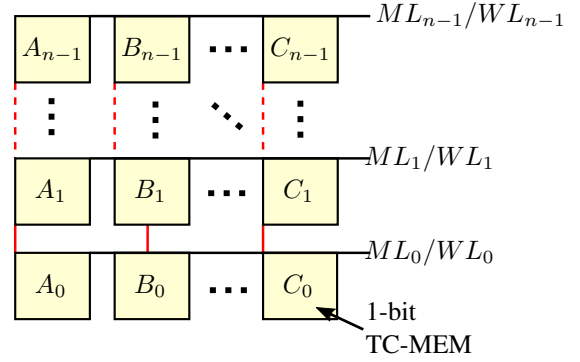


Figure 12: n -bit TC-MEM array simplified schematic.

A. Custom logical synthesis

The elementary logic gates presented in Sec. III can be integrated in logic synthesis environment [23] to create specific hardware accelerator that will keep one non-volatile operand. If the synthesized operation is integrate inside a complete computing system, it will lead to a reduction of the number of data transfers from memory to computing core.

B. Non-volatile reconfigurable circuit

FeFET based reconfigurable based circuit such as FPGA [15] can be created using non-volatile LUT presented in Sec. IV as well as small hardware accelerator that can be integrated directly inside computing system. Here, the advantage is that once the circuit is configured, there is no need to reconfigure it each time the circuit is powered, leading to increase the energy efficiency of the system. In addition, such concept can be applied to implement operation for IoT, AI and many other applications.

C. Reversible computing

Using TC-MEM, it is possible to leverage reversible computing for applications where some functions and their inverse need to be called. Let us consider a reversible function $f : x \rightarrow f(x)$. Storing the results $f(x)$ inside the TC-MEM array allows us to compute f and f^{-1} :

- In Memory mode, call the address x gives $f(x)$.
- In TCAM mode, search the word $y = f(x)$ gives the inverse function : $f^{-1} : y \rightarrow f^{-1}(y) = x$.

D. Partial word search

It is also possible to search parts of words stored inside the TC-MEM. Let us consider the word $A = A_{n-1} \dots A_1 A_0$ of the TC-MEM array presented in Fig. 12 and two integers x, y such as $0 < x < y < n - 1$. Three cases are possible:

- 1) The part $A_{n-1} A_{n-2} \dots A_y$ of A is searched: to do this, the remaining bits $A_{y-1} A_{y-2} \dots A_0$ of A have to be set in 'passing' don't care state.
- 2) The part $A_x A_{x-1} \dots A_0$ of A is searched: to do this, the remaining bits $A_{n-1} A_{n-2} \dots A_{x+1}$ of A can be set in 'passing' or 'blocking' don't care state.
- 3) The part $A_y A_{y-1} \dots A_x$ of A is searched: to do this, the less significant bits $A_x A_{x-1} \dots A_0$ of A have to be set

in 'passing' don't care state and the more significant $A_{n-1}A_{n-2}\dots A_{y+1}$ of A can be set in 'passing' or 'blocking' don't care state.

E. In memory computing

It is possible to directly use the the TC-MEM circuit in memory mode to achieve *And* operations. Indeed, calling two words at a time (A and B) from Fig. 12, we get Eq. 4:

$$\forall i \in [0; n - 1], WL_i = A_i \cdot B_i \quad (4)$$

In TCAM mode, if a value $X = X_{n-1} \dots X_1 X_0$ is presented to the address containing the word A , the generalization of Eq. 2 gives Eq. 5:

$$\forall i \in [0; n - 1], ML_i = \sum_{k=0}^i (A_k \oplus X_k) \quad (5)$$

Other logical functions such as binary *Xor* require to slightly modify the TC-MEM array.

VII. CONCLUSION AND PERSPECTIVES

In this article we presented an overview of three different LiM concepts offered by FeFET-based circuits. The presented circuits can be used to design specific operations with one non-volatile operand, to create fully reconfigurable circuit, and for in memory computing paradigm. All circuits have been designed using GLOBALFOUNDRIES 28nm technology and successfully tested on ASIC circuit.

Future work includes the use of the three different concepts to implement operation that we will integrate inside a processing system as hardware accelerators of directly inside the data path of a processor for example. A Hardware emulation platform based on FPGA will be created in order to test different applications such as cryptographic implementations and IoT data processing.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 780302 "3eFerro".

REFERENCES

- [1] H. S. Stone, "A logic-in-memory computer," *IEEE Transactions on Computers*, vol. C-19, no. 1, pp. 73–78, 1970.
- [2] N. Talati, R. Ben-Hur, N. Wald, A. Haj-Ali, J. Reuben, and S. Kvatinsky, *mMPU—A Real Processing-in-Memory Architecture to Combat the von Neumann Bottleneck*. Singapore: Springer Singapore, 2020, pp. 191–213.
- [3] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nature Electronics*, vol. 1, no. 6, pp. 333–343, 2018.
- [4] X. Chen, X. Yin, M. Niemier, and X. S. Hu, "Design and optimization of fefet-based crossbars for binary convolution neural networks," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2018, pp. 1205–1210.
- [5] Y. Zhang, L. Xu, K. Yang, Q. Dong, S. Jeloka, D. Blaauw, and D. Sylvester, "Recryptor: A reconfigurable in-memory cryptographic cortex-m0 processor for iot," in *2017 Symposium on VLSI Circuits*, 2017, pp. C264–C265.
- [6] M. Gao and C. Kozyrakis, "Hrl: Efficient and flexible reconfigurable logic for near-data processing," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2016, pp. 126–137.

- [7] W. Kang, L. Zhang, J. Klein, Y. Zhang, D. Ravelosona, and W. Zhao, "Reconfigurable codesign of stt-mram under process variations in deeply scaled technology," *IEEE Transactions on Electron Devices*, vol. 62, no. 6, pp. 1769–1777, 2015.
- [8] C. D. Wright, P. Hosseini, and J. A. V. Diosdado, "Beyond von-neumann computing with nanoscale phase-change memory devices," *Advanced Functional Materials*, vol. 23, no. 18, pp. 2248–2254, 2013. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adfm.201202383>
- [9] I. O'Connor, M. Cantan, C. Marchand, B. Vilquin, S. Slesazek, E. T. Breyer, H. Mulaosmanovic, T. Mikolajick, B. Giraud, J. 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)*, 2018, pp. 180–183.
- [10] A. Chen, "A review of emerging non-volatile memory (nvm) technologies and applications," *Solid-State Electronics*, vol. 125, pp. 25 – 38, 2016, extended papers selected from ESSDERC 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0038110116300867>
- [11] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie, "Pinatubo: A processing-in-memory architecture for binary bitwise operations in emerging non-volatile memories," in *Proceedings of the 53rd Annual Design Automation Conference*, ser. DAC '16. ACM, 2016, pp. 173:1–173:6. [Online]. Available: <http://doi.acm.org/10.1145/2897937.2898064>
- [12] E. T. Breyer, H. Mulaosmanovic, S. Slesazek, T. Mikolajick, and T. Mikolajick, "Demonstration of versatile nonvolatile logic gates in 28nm hkmf fet technology," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5.
- [13] X. Chen, S. Datta, X. S. Hu, M. Jerry, A. F. Laguna, K. Ni, M. Niemier, D. Reis, X. Sun, P. Wang, X. Yin, and S. Yu, "The impact of ferroelectric fets on digital and analog circuits and architectures," *IEEE Design Test*, pp. 1–1, 2019.
- [14] X. Yin, X. Chen, M. Niemier, and X. S. Hu, "Ferroelectric fets-based nonvolatile logic-in-memory circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 1, pp. 159–172, 2019.
- [15] X. Chen, K. Ni, M. T. Niemier, Y. Han, S. Datta, and X. S. Hu, "Power and area efficient fpga building blocks based on ferroelectric fets," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 5, pp. 1780–1793, 2019.
- [16] S. L. Miller and P. J. McWhorter, "Physics of the ferroelectric non-volatile memory field effect transistor," *Journal of Applied Physics*, vol. 72, no. 12, pp. 5999–6010, 1992.
- [17] X. Yin, K. Ni, D. Reis, S. Datta, M. Niemier, and X. S. Hu, "An ultradense 2fefet team design based on a multi-domain fet model," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 9, pp. 1577–1581, 2019.
- [18] E. T. Breyer, H. Mulaosmanovic, J. Trommer, T. Melde, S. Dünkel, M. Trentzsch, S. Beyer, S. Slesazek, and T. Mikolajick, "Compact fetfet circuit building blocks for fast and efficient nonvolatile logic-in-memory," *IEEE Journal of the Electron Devices Society*, vol. 8, pp. 748–756, 2020.
- [19] J. Li, R. K. Montoye, M. Ishii, and L. Chang, "1 mb 0.41 μm^2 2t-2r cell nonvolatile tcam with two-bit encoding and clocked self-referenced sensing," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 896–907, 2014.
- [20] S. Jeloka, N. B. Akesh, D. Sylvester, and D. Blaauw, "A 28 nm configurable memory (tcam/bcam/sram) using push-rule 6t bit cell enabling logic-in-memory," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 1009–1021, 2016.
- [21] K. Ni, X. Yin, A. F. Laguna, S. Joshi, S. Dünkel, M. Trentzsch, J. Müller, S. Beyer, M. Niemier, X. S. Hu, and S. Datta, "Ferroelectric ternary content-addressable memory for one-shot learning," *Nature Electronics*, vol. 2, no. 11, pp. 521–529, Nov. 2019.
- [22] K. Ni, B. Grisafe, W. Chakraborty, A. K. Saha, S. Dutta, M. Jerry, J. A. Smith, S. Gupta, and S. Datta, "In-memory computing primitive for sensor data fusion in 28 nm hkmf fetfet technology," in *2018 IEEE International Electron Devices Meeting (IEDM)*, 2018, pp. 16.1.1–16.1.4.
- [23] A. Bosio, M. Cantan, C. Marchand, I. O'Connor, P. Fiser, A. Poittevin, and M. Traiola, "Emerging technologies: Challenges and opportunities for logic synthesis," in *24th International Symposium on Design and Diagnostics of Electronic Circuits and Systems*, 2021.