POLITECNICO DI TORINO
Repository ISTITUZIONALE

Exploring the 3-D Integrability of Perpendicular Nanomagnet Logic Technology

Terms of use:

(Article begins on next page)

02 May 2024

Exploring the 3D Integrability of perpendicular Nano Magnet Logic Technology

Fabrizio Riente*† *Member, IEEE*, Daniel Melis*, Marco Vacca*

*Department of Electronics and Telecommunications, Politecnico di Torino, TO, I-10129 Italy

Corresponding author: †fabrizio.riente@polito.it

Abstract—Conventional integrated circuits' design uses one layer to place logic gates and many additional layers to route interconnections. This design technique is built around the constraints of MOSFET transistors. To further improve the performance of integrated circuits, it is necessary to go beyond this limitation and to design true 3D circuits. While this possibility is difficult to implement with transistor technology, perpendicular NanoMagnet Logic intrinsically enables the design of 3D devices. Its very low power consumption and the possibility to be integrated in the back-end of traditional fabrication processes. These characteristics make pNML an ideal candidate to implement low power coprocessors.

In this paper, we demonstrate the possibilities offered by pNML technology by designing a 3D coprocessor for the Summed Area Table, one of the most common algorithms used in image processing. We demonstrate the effectiveness of the design and the technology itself by comparing the performance with transistors implementations. The 3D design makes it possible to obtain a small circuit footprint. Overall the results presented here are a great step forward toward the design of 3D coprocessors in pNML technology.

Index Terms—Nano Magnetic Logic, pNML, 3D Architectures, Emerging Technologies

I. INTRODUCTION

Solid state electronics, and in particular, field effect transistors are the reason behind the extraordinary development of modern circuits. Two are the key factors that enabled such development, the scalability of the device and the possibility to use additional layers to route interconnection wires. The scalability of the device, ruled by the Moore's Law [**?**], has brought to a continuous scaling of the transistors' size, and a consequent improvement of performance and reduction of both power consumption and circuit area. The other key factor has been the possibility to use additional layers to route signals. The first layer is reserved to the transistors implementing the logic gates, while additional layers are used for interconnections. For example, the 14nm process of Intel allows up to 13 metal layers to route interconnections [**?**]. The use of so many additional wires has a huge advantage: transistors can be placed anywhere on the chip because they can be easily interconnected. This design process is called planar design, because the transistors are placed only on one single layer. Now that the scaling process is reaching the end [**?**], and so no further performance gain can be obtained, the only way to further improve circuits is to design new architectures based on improved components' layout.

The design of 3D circuits represents therefore a possibility to further improve integrated circuits. 3D devices can be obtained by also using MOS devices, using the through-silicon-via technology [**?**]. This technique is complex and it has many limitations [**?**]. In high density integrated circuits, it can cause routing congestion limiting its benifit [**?**]. There are also attempt to implement monolithically integrated 3D CMOS circuits, as reported in [**?**]. A true 3D circuit design implies the use of a technology that makes it possible to naturally stack logic gates, enabling vertical 3D integration. Few emerging technologies make this possibility real, for example Skybridge [**?**] [**?**], enables the design of 3D circuits. Perpendicular NanoMagnet Logic (pNML) naturally enables the design monolithically integrated 3D circuits. Among the non-charge-based technologies, NanoMagnet Logic is considered a promising option [**?**]. In this technology, the basic element is a nanomagnet with two stable configurations [**?**] [**?**] [**?**]. It behaves as a logic element by interacting with the stray field of neighboring magnets. It propagates the information through domain wall motion, acting also as an interconnection element. Finally, due to its magnetic nature, nanomagnets behave has memory devices. Given that, the information is computed and propagated through field coupling among neighboring elements. Magnets can be organized on the same physical plane or on different planes. In both cases, the information propagates correctly as experimentally demonstrated in [**?**]. As a consequence, the use of this technology makes it possible to break the bonds of the traditional integrated circuit design process. Here, logic, memory and interconnection elements are intrinsically embedded within the same device and can be placed on any layers. Furthermore, this technology can be integrated in the backend of standard fabrication processes, leading to hybrid circuits CMOS/pNML. The exchange of the information among pNML/CMOS can be guaranteed by Magnetic Tunnel Junction (MTJ) devices, properly organized at the interfaces. The pNML circuit can be used to implement dedicated coprocessors physically fabricated on the interconnection layers.

The goal of this work is to demonstrate the potential offered by this new design methodology and the technology itself. A complex circuit has been designed exploiting as much as possible the potential offered by the third dimension. The circuit is a dedicated accelerator for the Summed Area Table (SAT) algorithm, an algorithm commonly employed in image processing. The SAT algorithm has been chosen for its modularity. This algorithm operates on a matrix of pixels, so hardware accelerators can be designed exploiting a regular spatial organization. The circuit is divided in processing ele-

ments, each of them is a multilayer structure with a logic part and a memory part, thus implementing a Logic-In-Memory structure. Furthermore, this algorithm is particularly representative for a class of algorithms that operates on a matrix. In particular, calculation are performed on neighboring elements of the matrix. In our opinion, it is therefore particularly useful for highlighting the benefit of a 3D circuit design.

To summarize, this work has several major contributions:

- We propose a 3D design of a coprocessor for image processing based on pNML technology as case study.
- The performance of the coprocessor have been carefully evaluated and compared against the same circuit implemented on CMOS technology. The results obtained clearly highlight a relevant gain in terms of area and power consumption, despite the pNML is on a much larger technology node. Running the circuits at the same frequency it is possible to observe an evident static power contribution within CMOS designs. Results also highlight a much lower speed, when comparing pNML with CMOS. Thus, this technology is targeted for low power and low speed applications.

II. BACKGROUND

In perpendicular Nano Magnet Logic (pNML) nanomagnets are characterized by a perpendicular magnetic anisotropy (PMA). This magnetic property is obtained with a multilayer stack of Co/Pt or Co/Ni. Ferromagnetic domains are defined by patterning the PMA stack on top of a Si substrate [?]. Two stable magnetization states can be identified, both perpendicular to the substrate plane [?]. The binary information is encoded into these two stable configurations of the ferromagnetic islands (figure ??.A). In this technology, crystalline and interfacial anisotropy prevail, enabling the design of magnets with different shapes (circular, squared, elongated stripes, etc...). Magnetic coupling is exploited to perform logic computation and propagate the digital information in pNML circuits. However, two neighboring equal-shaped nanomagnets experience the same coupling leading to an undefined propagation direction of the information. To overcome this issue, a focused ion beam irradiation, with Ga^+ ions, is applied on a specific spot of the magnet (figure ??.C). This irradiation changes locally the magnetic anisotropy of the device, making it more sensitive to magnetic fields. The area where the anisotropy is locally reduced is named artificial nucleation center (ANC) [?]. It defines the region where the magnetization reversal starts, i.e. where the domain wall nucleates [?].

To control the signal propagation in pNML circuits a global, uniform, out-of-plane clock field is applied over the whole circuit [?]. The schematic representation is depicted in figure ??.B. The principle behind the signal propagation is depicted in figure ??.E. Each magnet is mostly influenced by its next left neighbor. Figure ??.E, at time t=0, shows an initial configuration where no external field is applied. At t=1, a positive pulse is applied, the coupling field of the input magnet (M1) superposes the applied clock field enabling/preventing the switching of the cell M2. Neighboring cells tries to reach
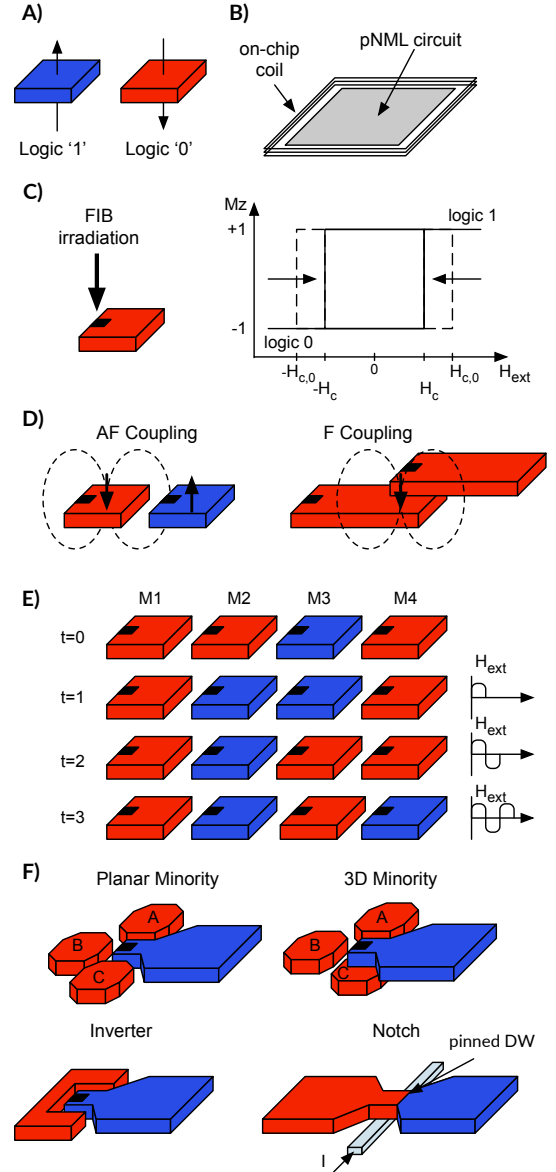

Figure 1: A) pNML elementary cells; B) Schematic representation of pNML computing system; C) Coupling among neighboring cells lying on the same physical plane or the subsequent layer; D) The FIB irradiation lowers locally the PMA creating the ANC; E) Signal propagation on a chain of nanomagnets when an external out-of-plane field is applied; F) Basic gates like minority (planar and 3D), inverter and the notch as synchronization element.

the low energy state, aligning their magnetization in an anti-parallel/parallel configuration. Indeed, magnets that lie on the same plane are anti-ferromagnetically (AF) coupled, whereas nanomagnets placed one above the other are ferromagnetically (F) coupled. pNML technology makes it possible to design 3D circuits (??.C). Monolithically integrated 3D devices have been experimentally demonstrated in [?] [?] [?]. The possibility to exploit the third dimension gives more freedom to the designer. The basic gates available are the inverter (figure ??.F), which is simply obtained by cascading two
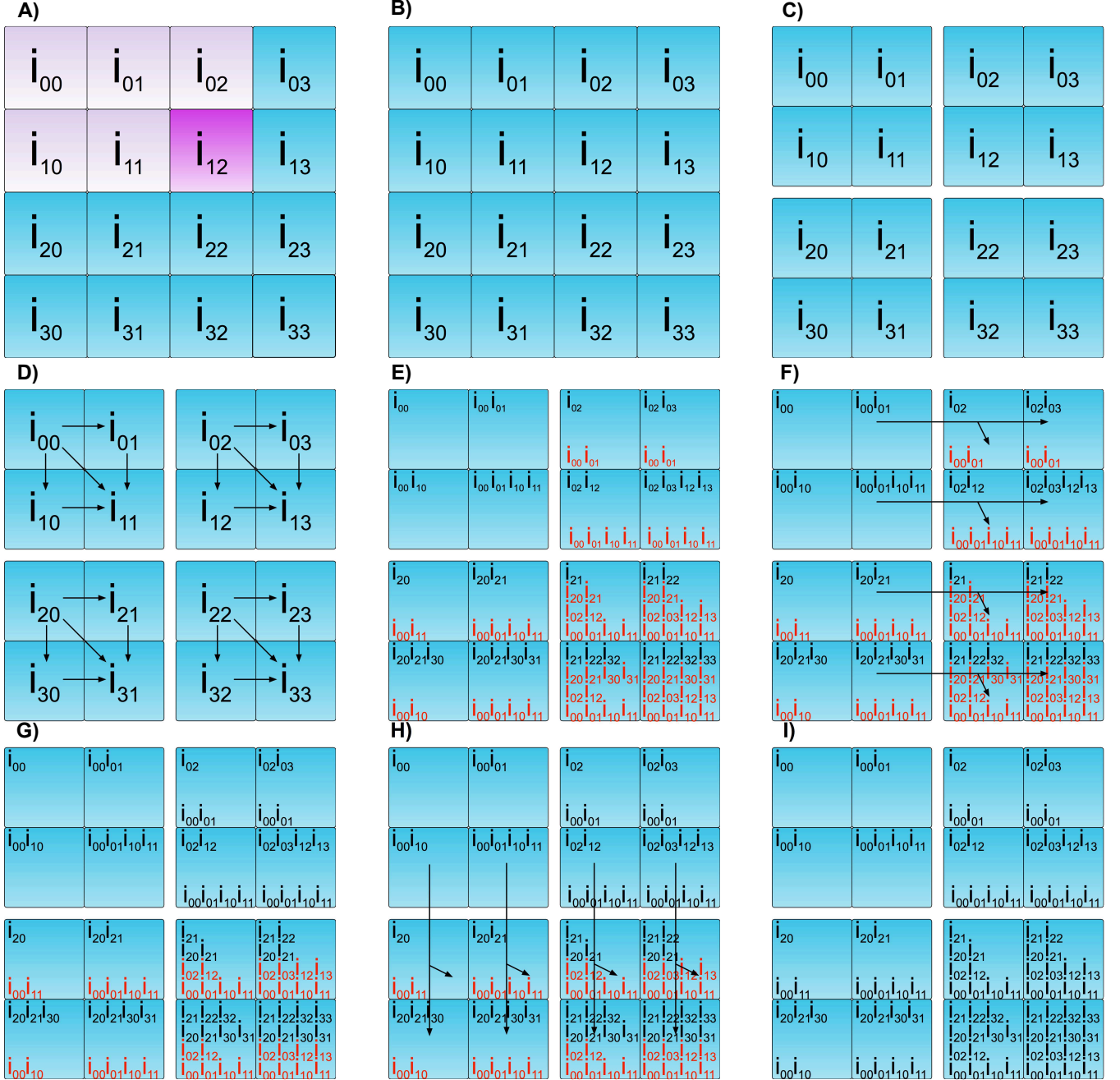
Figure 2: Detailed steps of the SAT algorithm. Arrows represent sums. The elements in black are the already stored in the memory location (all summed with each other), the red ones are the values missing to reach the final value of the algorithm. A) Elements involved in the calculation of the SAT in position (1, 2); B) Matrix to be processed; C) Matrix sub-division; D) Local execution of SAT algorithm; E) Results of the previous step; F) Sharing of partial results among sub-matrices; G) Results of the previous step; H) Sharing of partial results among sub-matrices; I) Algorithm completed.

elements, and the 3-input minority voter (figure **??**.F). The latter can also be used as programmable gate. Indeed, by fixing one of the input to logic 0/1 it is possible to obtain the NAND/NOR function respectively. Both planar and 3D implementation have been experimentally demonstrated [**?**]. Another fundamental element is the notch, which is used for signal synchronization [**?**]. Here, the incoming domain wall is pinned at the geometrical deformation of the magnetic wire

(figure **??**.F). The notch creates an energy barrier that pins the incoming information [**?**]. The propagation can be restored by applied a current pulse through a wire buried in the substrate, as experimentally demonstrated in [**?**] . The pulsed current generates an in-plane field that superposes with the out-of-plane field, providing sufficient energy to release the stuck domain wall.

III. SAT ALGORITHM

In this section a brief description of the Summed Area Table (SAT) algorithm is given to explain the main execution steps, fundamental to understand to pNML hardware implementation. It is mainly used in computer graphics applications [?]. It takes as input a two-dimensional array of numerical values, and computes a new matrix where the calculated element at any point (x,y) is the sum of all values in a rectangular subset of the array. From a mathematical point of view, it can be expressed as:

$$I(x,y) = \sum_{\substack{x'<x \\ y'<y}} i(x',y')$$

The SAT in a given position (x,y) is the sum of all elements above and to the left of the point (x,y) inclusive (figure **??**.A). Thus, the SAT value of a given position is strictly related to the values in the neighboring locations, suggesting many possible optimizations to speed up the execution. For example, the array could be splitted into sub-matrices of size $2^i \text{x} 2^i$, where the algorithm is carried out locally (namely, considering the sub-matrix as a single, isolated matrix). The calculation in every sub-matrice is performed independently and in parallel. At the subsequent time step (i+1), data from the consecutive arrays of size 2^{i+1} are passed from a sub-matrix to the other, sparing the effort to compute again some of the sums. Thus, partial SAT are computed in every sub-array. Figure **??** shows some execution step for a 4 by 4 matrix, the largest one processed in this work. Figure **??**.D represent the local execution of the SAT on a 2x2 matrix. The text reported in black represents values already computated and available in a given location (figure **??**.E). Red text instead, indicates elements that are still missing to obtain the final SAT. The red values should be added to compute the final SAT (figure **??**.I).

IV. CIRCUIT DESCRIPTION

As mentioned in section **??**, the array is divided into sub-matrices to speed-up the computation. This sub-array identifies the basic computational element of the circuit, that from now on is called cell. A cell is able to process a 2x2 matrix, and can interact with neighboring cells to process a larger array. The design here proposed tries to exploit as much as possible the 3D integrability of pNML technology. The whole design of the cell has been done by using the tool MagCAD [?], which is part of the ToPoliNano framework [?]. It embeds a physical compact model of the technology presented in [?]. The circuit performance are extracted through VHDL simulations by exploiting the circuit netslit extracted by the used software.
The single cell schematic representation is depicted in figure **??**. It consists of two parts:

- The control, hosted in the top layers, is composed of four Finite State Machines (FSMs);
- The datapath, in the bottom layers. It embeds four memory locations and an adder accumulator;

The logical separation among control and datapath well suits the Logic-In-Memory concept. The design of the single cell reported in the following, works on a data parallelism of 4
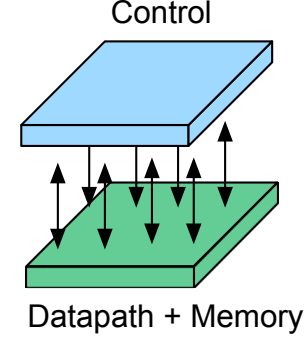

Figure 3: Schematic representation of the basic cell architecture.
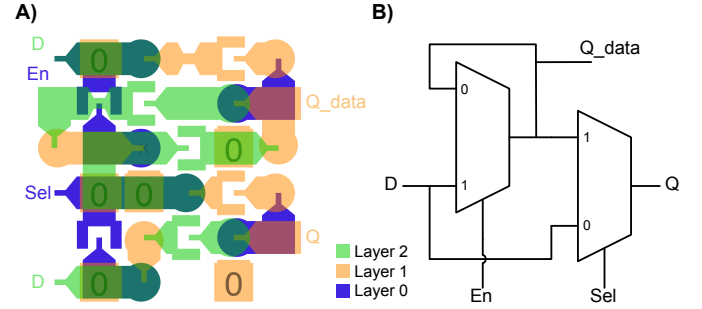

Figure 4: A) pNML layout of 1-bit memory cell; B) Its schematic representation

bits. The rest of the section includes a brief description of the most important elements of both the datapath and the control.

A. Datapath

1) Memory: The memory is the wider element, on top of which are placed the FSMs of the control section. The schematic representation of the 1-bit memory cell is shown in figure **??**.B. Its corresponding pNML layout is reported in figure **??**.A. It is based on the implementation presented in [?], here optimized in term of area occupation.
Magnetic materials can intrinsically retain the binary information over time on the magnetization vector. However, here a feedback is implemented to decide when the information should be stored.
It consists of two cascaded multiplexers; in one of them, the output signal is brought back to one of the inputs. This implements the actual memory loop. The other multiplexer switches between the stored element and the external datum.

Inputs		Outputs	
Sel	Enable	Q	Q_data
0	0	D	*Stored Datum*
[1] 0	1	D	D
1	0	*Stored Datum*	*Stored Datum*
1	1	*Stored Datum*	D

[1] In practical use, this is a forbidden input configuration
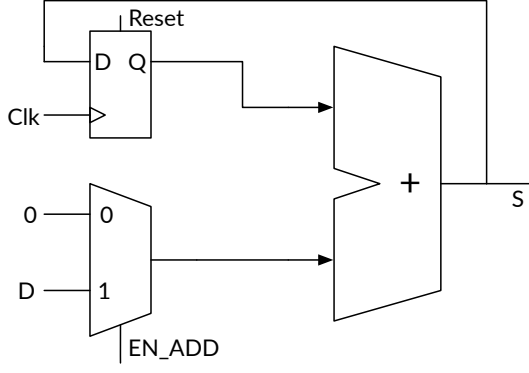
Table I: Logical behavior of the memory cell

Figure 5: Circuit schematic of the Adder-Accumulator.

Figure 6: The signals of a basic cell

To sum up: if the datum is to be stored into the current location, the memory element opens and stores it, otherwise the datum is passed to the next memory location. The truth table of this circuit is shown in table **??**. The memory is controlled by means of a decoder and a write-read signal. The input datum might come from the outside (at the beginning, when the matrix to be processed is being loaded) or from the output of the adder accumulator, described in the next section. Each single cell involved in the local SAT calculation contains four memory locations.

2) Adder-Accumulator: In order to keep the design simple, the adder chosen is a ripple carry adder. It has been implemented exploiting the minority gate available in pNML. This component takes as one of the inputs the datum read from the memory. It stores the cumulative sum, or, alternatively, can be reset to start a new computation discarding the previously stored value. The schematic representation of the proposed circuit is reported in figure **??**.

3) Datapath Signals: The set of control signal of the datapath are the following ones:

- R_W: distinguishes the reading or writing operation on the memory;
- SEL: switches the input port of the memory between the data coming from outside and the data coming from the adder-accumulator;
- *Add_0*, *Add_1*: the address of the memory location;
- EN_ADD: determines if the current output datum of the memory is to be brought in input to the adder-accumulator;
- $RESET$: resets the operation chain of the adder-accumulator. It is used when a new addition has to be started.

These signals are schematically represented in figure **??**. They are all driven by the control section. The only signals that have to be provided from the outside to the datapath section, with the proper timing, are the data signals.

B. Control

The control is actually made with four separated and autonomous FSMs. They all have active high output signals, so that they can be activated one at a time. Their outputs are in
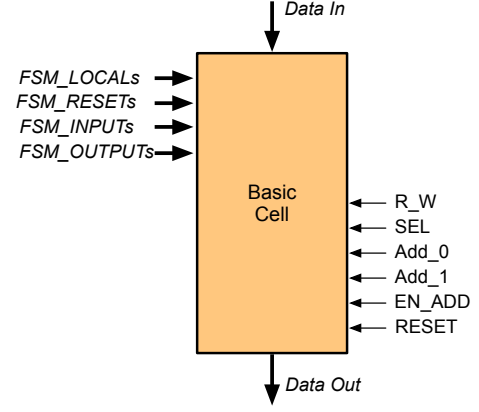
OR and just the active one drives a logic one.
Each FSM has a particular task to perform; these are:

- Reset FSM: loads the data to be processed into the memory;
- Local FSM: carries out the SAT algorithm among the four locations of the cell;
- Input FSM: adds the data coming for a neighboring cell into the proper locations;
- Output FSM: outputs the data to be sent towards a neighboring cell.

Input and output FSMs work for the data transfer both in the vertical and the horizontal direction. They have a specific signal to distinguish the two cases. Besides this signal, all the FSMs have only an input, the active-high reset: each FSM starts as soon as this signal is pulled down, and stops as soon as it is pulled up again.

If a 2x2 array is considered, a single cell is sufficient to compute the SAT. In this particular case, just the *Reset* and *Local* FSM are enough to perform the calculation.

However, the case study considered has four cells arranged in a 2x2 structure in order to process a 4x4 array of numerical values. In this case, the cells can work independently when computing the SAT, but some synchronization is required when data should be exchanged. The order in which the FSMs are triggered is:

- Reset FSM, for all the cells;
- Local FSM, for all the cells (figure **??**.D);
- Input FSM for North-East and South-East cell, output FSM for North-West and South-West cell (everything at the same time, figure **??**.F) ;
- Input FSM for South-West and South-East cell, output FSM for North-West and North-East cell (everything at the same time, figure **??**.H)

A global FSM triggers the right FSM in the right cell with the proper timing.

The design contains several pipeline stages, whose main purpose is simplifying the design and the debug. It is important to keep in mind that this limits the final performance, in particular the circuit latency.

The 3D pNML feature is very useful to achieve a compact design. As mentioned above, the basic cell can process a
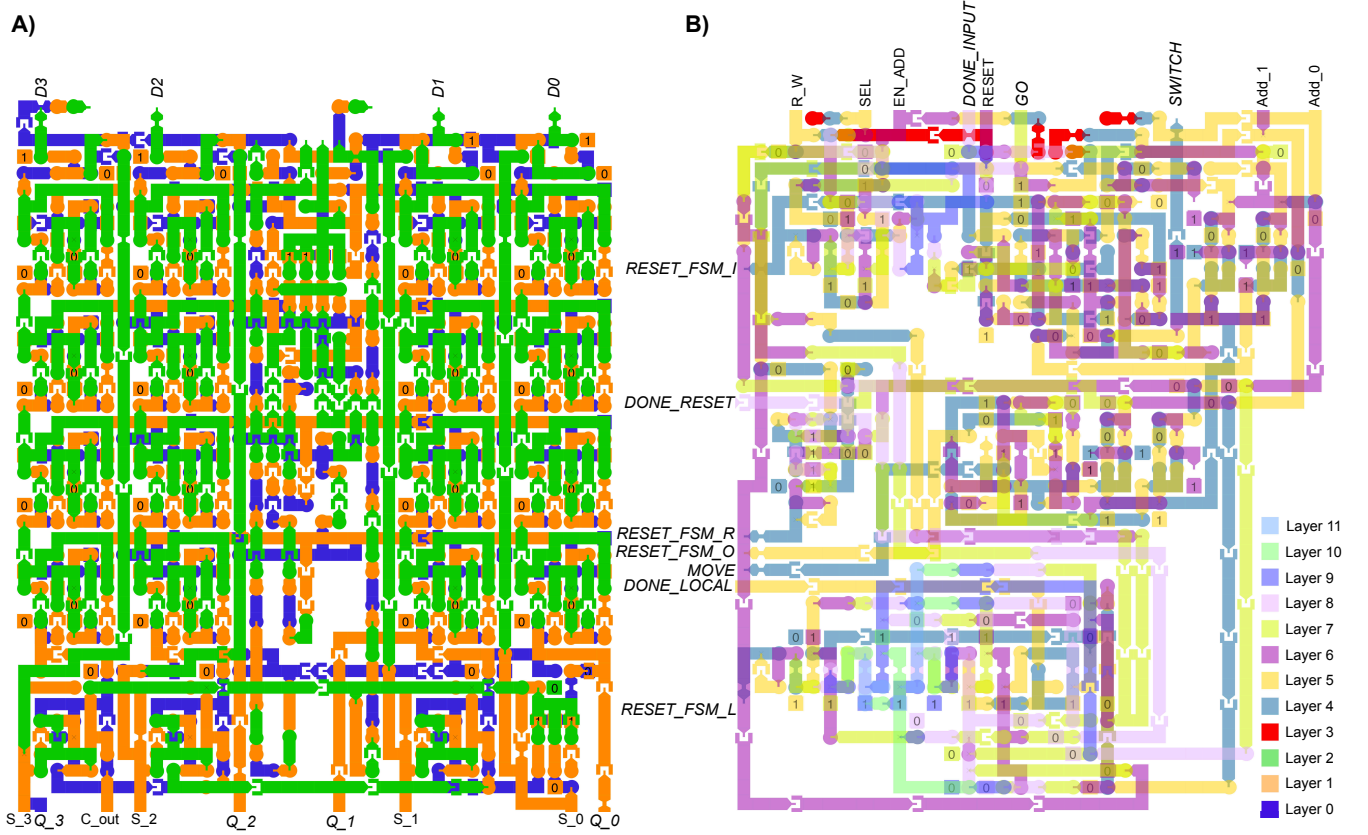
Figure 7: pNML layout of the basic cell: A) Datapath; B) Control.

2 by 2 matrix. We designed it in pNML on 12 physical layers. The first three layers (0-2) host the memory array along with the adder-accumulator; the layer just above is a routing layer, and the other layers on the top house the four FSMs machines. Figure **??** shows the datapath (**??**.A) and the control section (**??**.B); they are stacked one on top of the other. The input/output signals of this fundamental cell are shown in figure **??**, where the ones not in italic are signals that come from the control to the datapath, and are unaccessible from the outside.

Four basic cells are arranged in a 2 by 2 array: as a whole, they can process a 4 by 4 matrix. This arrangement needs two more layers for routing and hosting the global FSM, which makes 14 the total number of layers. Since every basic cell can work independently from the other, a larger computational array can be implemented by changing the design of the global FSM.

V. PERFORMANCE

The pNML architecture presented in the previous section has been designed by using MagCAD tool [**?**]. The correct circuit's behavior has been verified through VHDL simulations. We have exploited MagCAD to extract the VHDL netlist associated to the designed circuit. The extracted netlist embeds a compact model of the technology verified through experiments [**?**] [**?**]. Besides the verification, we have compared the pNML architecture with two CMOS equivalent designs. The first version is an ASIC CMOS implementation, named "ASIC

CMOS" in table **??** and **??**. The second version is a CMOS architecture that mirror the pNML design, named "pNML-like" in table **??** and **??**. This implementation mimics the structure of the magnetic circuits, with all the pipeline stages intrinsically present in such designs. For example, in pNML, the memory cell is made with the logical scheme shown in figure **??**.B, which is not common in CMOS designs. All these architectures have been synthesized both with a nandgate $45\,\mathrm{nm}$ library and with a $28\,\mathrm{nm}$ library with a timing constraint of $1\,\mu\mathrm{s}$. The chosen timing constraint is met for all CMOS implementations.

On the other hand, the pNML implementation considers nanomagnets whose typical width is $220\,\mathrm{nm}$ and magnet inter-space of $160\,\mathrm{nm}$. Those dimensions are typically used in experiments to obtain good measurements. However, the projection of a scaled implementation of the technology with nanowire width of $90\,\mathrm{nm}$ and magnet inter-space of $80\,\mathrm{nm}$ is reported. Authors in [**?**] show that it possible to obtain Co/Pt nanomagnets with a switching field lower than 25mT. Therefore, for minimum power dissipation we have considered a clocking field of 20mT, generated with the on-chip inductor presented in [**?**]. The authors show that for operating frequencies below 20MHz, both the hysteresis and Eddy current losses within the cladding material are negligible. The main contribution come from resitive losses. From the VHDL simulations, is it possible to observe that the maximum operating frequency is below 1MHz. This number is obtained from the minimum pulse width, which is evaluated as the sum of the propagation

	ASIC CMOS 28 nm	ASIC CMOS 45 nm	CMOS pNML-like 28 nm	CMOS pNML-like 45 nm	pNML 220 nm	pNML 90 nm
Clock Period	1 μs	1 μs	1 μs	1 μs	1.676 μs	1.545 μs
Total Time	1 μs	1 μs	19 μs	19 μs	923 μs	851 μs
Throughput	1 MOps	1 MOps	52.63 KOps	52.63 KOps	1.08 KOps	1.17 KOps
$\left(\frac{Throughput_{pNML@220}}{Throughput_{CMOS}}\right)$	1.08 $\cdot 10^{-3}$	1.08 $\cdot 10^{-3}$	20.52 $\cdot 10^{-3}$	20.52 $\cdot 10^{-3}$	-	
Area	132 μm^2	217 μm^2	281 μm^2	466 μm^2	169 μm^2	54 μm^2
$\left(\frac{Area_{pNML@220}}{Area_{CMOS}}\right)$	1.28	0.78	0.60	0.36	-	-
Static Power	1.11 μW	4.20 μW	2.64 μW	9.38 μW	-	-
Dynamic Power	0.21 μW	0.47 μW	0.26 μW	0.52 μW	2.87 μW	0.92 μW
Total Power	1.32 μW	4.67 μW	2.90 μW	9.90 μW	2.87 μW	0.92μW

Ops= Operations per second; KOps= Kilo Operations per second; MOps= Mega Operations per second; GOps= Giga Operations per second

Table II: Delay and area comparison between the pNML architecture and the CMOS one, with a 2 x 2 SAT size

	ASIC CMOS 28 nm	ASIC CMOS 45 nm	CMOS pNML-like 28 nm	CMOS pNML-like 45 nm	pNML 220 nm	pNML 90 nm
Clock Period	1 μs	1 μs	1 μs	1 μs	1.750 μs	1.597 μs
Total Time	2 μs	2 μs	67 μs	67 μs	3.4 ms	3.1 ms
Throughput	500 KOps	500 KOps	14.92 KOps	14.92 KOps	294 Ops	322 Ops
$\left(\frac{Throughput_{pNML@220}}{Throughput_{CMOS}}\right)$	0.58 $\cdot 10^{-3}$	0.58 $\cdot 10^{-3}$	19.70 $\cdot 10^{-3}$	19.70 $\cdot 10^{-3}$	-	
Area	933 μm^2	1523 μm^2	2226 μm^2	3761 μm^2	1249 μm^2	401 μm^2
$\left(\frac{Area_{pNML@220}}{Area_{CMOS}}\right)$	1.34	0.82	0.56	0.33	-	-
Static Power	8.03 μW	30.63 μW	19.76 μW	80.12 μW	-	-
Dynamic Power	1.41 μW	3.43 μW	1.66 μW	2.68 μW	21.20 μW	6.81 μW
Total Power	9.44 μW	34.06 μW	21.42 μW	82.80 μW	21.20 μW	6.81 μW

Ops= Operations per second; KOps= Kilo Operations per second; MOps= Mega Operations per second; GOps= Giga Operations per second

Table III: Delay and area comparison between the pNML architecture and the CMOS one, with a 4 x 4 SAT size

time within the nanowire (t_{prop}), the domain wall nucleation time (t_{nuc}) and the rise time. In particular, the pulse rise time is estimated as 25% of $t_{prop}+t_{nuc}$ as reported in [**?**], which is a conservative assumption on pulses within the micro-second range. Thus, considering the obtained working frequency we can neglect cladding losses. From the analysis in [**?**], the power density is about $1.7\,\mathrm{W/cm}^2$ when clocking the circuit at 1MHz. Therefore, this power density can be considered a good approximation for our circuits. To make the comparison as fair as possible, we have considered a clock frequency of 1MHz also for all the CMOS implementations. The outcomes of this comparison are listed in tables **??** and **??**. The pNML implementation with a nanowire width of $220\,\mathrm{nm}$ has been considered as reference.

The metrics analyzed are the clock frequency, the total computation time, the area occupation and the power consumption. We have applied 1MHz clock frequency to all CMOS circuits. The ASIC CMOS computation time is one clock cycle when considering the 2x2 array and two clock cycles with the 4x4 array. The pNML-like CMOS architectures, which mirror the pNML behavior, show a higher computation time due to the pipeline stages intrinsically present those circuits. Tables **??** and **??** show, as expected, that the CMOS implementation has a higher throughput than the pNML one, by almost three orders of magnitude. The throughput is expressed as number of computation per seconds. However, this design is not performance oriented, but rather just considers its feasibility from an architectural point of view. Hence, some optimizations could

speed up the pNML performance. Another reason accounts for the high delay of the pNML circuit. In the model, it has been considered a "worst case" scenario: a value propagating from a layer to the one above or below has to wait a clock cycle. This is a safe assumption, and it simplifies the design. However, from the physical point of view it is incorrect: the magnetization vector can propagate through the different layers without the need of a new clock pulse. The design is full of signals moving across many layers, and the model represents them as having a delay larger than the real one. Last, the delay strongly depends on the physical parameters of the technology (wire sizes, magnetic field intensity, and so on). Due to its early stage of development we have considered the standard values measured from experiments. Thus, there is no optimization on this side. In addition, the compact model behind the VHDL simulations considers a rise time that is 25% of the pulse width [**?**] [**?**].

Area, on the other hand, is larger in the CMOS versions. Nevertheless, the ratio is not as great as it was for delay, being the pNML circuit as large as about 80% of the $45\,\mathrm{nm}$ ASIC version's area. This compactness in term of area is mainly due to the several layers used in the design (14 layers for the 4 x 4 circuit, 12 for the 2 x 2). The $90\,\mathrm{nm}$ pNML prediction shows further improvements in term of area when compared to the $28\,\mathrm{nm}$ ASIC implementation. In this case, about 60% less area is required.

With a clock frequency of $1\,\mathrm{MHz}$ it is possible to observe that the main power contribution in CMOS designs is related

to static losses. The 220 nm pNML show a higher power consumption when compared to the 28 nm ASIC implementation. However, the power losses goes down by 30% in the 2x2 array, when the 90 nm pNML is considered. The gain is even higher, about 80%, if compared to the 45 nm ASIC implementation. These results highlights the main advantage of pNML technology, the absence of static energy consumption, coupled with the low dynamic power consumption. The absence of static energy consumption in particular is a considerable advantage, because it allows to greatly reduce energy consumption by keeping the circuit in idle while it is not used.

VI. Conclusions

In this paper, we have presented the design of a complex architecture by using the pNML technology as target. We exploited the monolithic 3D integrability of the technology to increase the circuit compactness. We implemented and compared the ASIC and pNML-like CMOS architectures to their pNML counterpart. The results show that there is a remarkable static power consumption in CMOS designs. We showed that the pNML technology offers a possible solution for reducing energy consumption, by having at the same time a low power consumption and no static power consumption at all. It is slower, but there is room for improvement. We think it worth to continue investigating this technology to improve it. Faster domain wall nucleation and domain wall propagation can lead to better pNML timing performance, increasing the both the final throughput and the minimum clock frequency. Moreover, it is not easy to compare two completely different technologies. In particular, CMOS that is a well-known and consolidated technology and pNML that is at early stage of development.

M. Vacca received the Dr.Eng. degree in electronics engineering and the Ph.D. degree in electronics and communications engineering from the Politecnico di Torino, Turin, Italy, in 2008 and 2013, respectively. He is currently a Research Assistant at the Politecnico di Torino. His research interests include nanomagnet logic and others beyond-CMOS technologies. He is also an expert of innovative and unconventional computer architectures.

F. Riente received his M.Sc. Degree with honors (Magna Cum Laude) in Electronic Engineering in 2012 and the Ph.D. degree in 2016 from the Politecnico di Torino. He was Postdoctoral Research Associate at the Technical University of Munich in 2016. He is currently Postdoctoral Research Associate at the Politecnico di Torino. His primary research interests are device modeling, circuit design for nano-computing, with particular interest on magnetic QCA. His interests cover also the development of EDA tool for beyond-CMOS technologies, with the main focus on the physical design.

D. Melis received his M.Sc. in Electronic Engineering in 2018 from Politecnico di Torino. Ever since his master thesis work, his research interests have involved beyond-CMOS technologies (specifically, perpendicular Nano Magnetic Logic ones), both from the layout and from the architectural point of view.