Large scale, high density integration of All Spin Logic

Qi AN
C2N, Université Paris-Sud
UMR CNRS 9001,
Orsay, France
anqi_91@sina.com

Sébastien Le Beux,
INL,
Ecole Centrale de Lyon
Lyon, France
sebastien.le-beux@ec-lyon.fr

Ian O'Connor
INL
Ecole Centrale de Lyon
Lyon, France
ian.oconnor@ec-lyon.fr

Jacques-Olivier Klein
C2N, Université Paris-Sud
UMR CNRS 9001,
Orsay, France
jacques-olivier.klein@u-psud.fr

***Abstract*—**Spintronics brings new features that make it a viable candidate technology to implement non-conventional processing for new computing paradigms in an efficient way. The first milestone of the spintronics roadmap was the fabrication of hybrid systems where the data processing relies mostly on charge-based electronics devices (CMOS), while the memory hierarchy is partially or totally replaced by MRAM. In the next step, spintronics can also be used for data processing, still in conjunction with CMOS. Nevertheless, replacing all the processing by pure spintronic circuits, without any charge current, remains the ultimate objective of spintronics. All spin logic (ASL) paves the way towards that goal, even if some CMOS control circuits are still necessary. However, as ASL does not rely on the same computing principle as CMOS, it is necessary to address some specific issues. Pure spin current propagates in every direction, including backwards in the presence of multiple inputs; and is divided when crossings are encountered. It combines mainly linearly, while logic operations require non-linear binary decisions. Interconnect between logic gates requires directionality from inputs to outputs, and fanout with negligible signal attenuation. In this context, we develop new strategies for ASL modeling and logic design. We propose an architecture and a design strategy based on a high-density array to address the specific issues of directionality, attenuation and linearity. Moreover, the feasibility is supported through the modeling and the simulation of its basic block. This implies modularity to simulate complex circuits, even when they are ahead of today's experimental demonstrations.

Keywords—All spin logic, modeling, array architecture.

I. Introduction

Information is traditionally represented by electrical quantities: historically current [1], but now voltage, essentially because voltage spreads along interconnections while current is divided when crossing fanout elements. Consequently, the modularity of a computing circuit relies on the interconnect of logic gates through an asymmetric policy: a single output per logic gate is connected to multiple inputs of the following logic stage. In CMOS design, input and output logic states correspond to the charge of the gate capacitors. Channel miniaturization of CMOS transistors leads to increased static power consumption due to leakage, while data access, especially during reading to and writing from memories, is responsible for much more energy consumption than data processing. In this context, some emerging technologies offer the opportunity to challenge these limits. First, by mixing memory and computing the read/write operation is greatly reduced. In addition, the use of non-volatile circuitry eliminates leakage in standby mode. Ultimately, the objective is to switch from charge current to another physical quantity to carry information: magnetic polarization. This constitutes the birth of spintronics [2,3]. The first generation of spintronics architectures were based on magnetic tunnel junctions (MTJ) as nonvolatile memories. Moreover, MTJs have been considered as possible computing elements involved in calculation in the context of memory-in-logic architectures. With magnetic domain wall (MDL) pushed by magnetic fields or by polarized current, spintronics is on the roadmap of the total replacement of CMOS transistors, for storage and for calculation applications. More recently, experiments have shown non-local effects with pure spin currents. We then approach the ultimate objective: completely eliminate charge-current and associated power consumption through the Joule effect.

This work focuses on the design methods that can be adapted to such "all-spin-logic" architectures.

Conventional methods for logic synthesis usually depend intrinsically on the mode of operation of functions implemented with CMOS transistors. More than a current switch, the CMOS transistor acts as a voltage-controlled connector between the single output of the logic gate and one of the two fixed voltage sources representing the two logic states (VDD or GND). This leads to a total asymmetry between the low impedance of the power supply and the high impedance of the input of the next block. This asymmetry provokes the propagation of the signal in one direction, from the output of a logic block, to one or more inputs of the following blocks, and never the opposite.

Operating with an ASL block is totally different. A spin current is generated thanks to a bipolar charge current in a magnetic polarized layer, typically a fixed layer part of a MTJ. This polarization propagates through the channel to one (or a limited number of) outputs, typically by switching its magnetic direction. The output magnetic state results from a linear superposition of the effect of multiple inputs, and the logic function typically implements a threshold logic with a combination of majority gates. The information is carried by a spin current. As a consequence, when crossing another channel, be it connected to another input or another output, the current forks into each branch, including backward to the other inputs.

978-3-9819263-0-9/DATE18/©2018 EDAA

In addition, in ASL, the channels are considered simultaneously as a computing element and interconnect. This new intrication of interconnect and computing, along with the lack of directionality and the prevalence of the majority gate in logic synthesis makes it necessary to completely rethink computing principles for ASL. Thus, CMOS logic design methods are not directly usable. In this context, our contribution is twofold. First, we consider a high density computing architecture that benefits from the specific features of ASL through drastically reduced interconnect overhead by abutting replicated basic blocks. The feasibility of this architecture is supported through a modular modeling scheme that allows demonstrate the operation of our basic block to be demonstrated.

Section 2 will explain why the ASL operation principle requires new computing architectures. A high density ASL array architecture that reduces the interconnect overhead is presented in section 3. Section 4 is dedicated to modeling techniques that are implemented in section 5 to simulate the basic block of our ASL array.

II. ASL OPERATION PRINCIPLE REQUIRES NEW COMPUTING ARCHITECTURE

A first example of ASL circuit considered in our paper is shown in Figure 1 (a). It corresponds to "copy" and "not" functions. This circuit is composed of two perpendicular MTJs[4,5] and a spin transport channel. Underneath the MTJs, a spacer is placed in the middle, dividing the MTJ into injector/detector, so MTJs are used alternately as input (injector) and output (detector). A tunnel barrier is placed between the injector MTJ and the channel to enhance the spin injection efficiency [6,7], whereas the interface between the detector MTJ and channel is transparent. A ground lead is placed near the injector to guarantee the non-reciprocity of the ASL device [8].

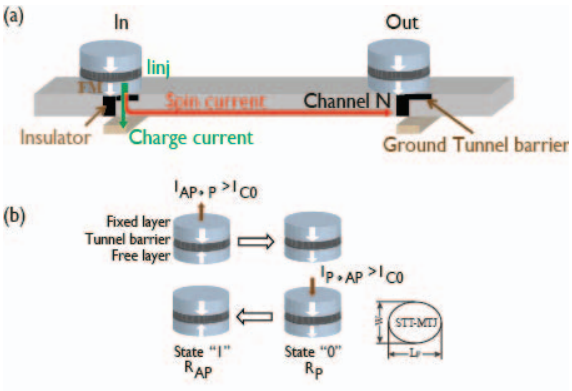

Fig. 1. ASL circuit components, with channel and MTJ as spin current injector and detector.

The MTJ in Figure 1 (b) is composed of two ferromagnetic (FM) layers isolated by an oxide layer. Depending on the relative magnetization orientations of the two FM layers-which influences the MTJ resistances, one MTJ can represent two different states: "0" or "1". The state of MTJ can be changed by applying a voltage/current *Vwrite*/*Iwrite* above the critical current I_{c0}. After writing the MTJ state S_{in}\{0, 1\}, a charge current I_{inj} is injected to ASL channel, producing a polarized spin current. The spin current is then transported to the detector MTJ through the channel. With spin-flipping, the spin current in the detector I_{sdet}, will switch the MTJ state if it is larger than the critical current I_{c0}. The switched state *Sout* depends on the injection charge current I_{inj} polarity and the injector MTJ state S_{in}. If I_{inj} is negative, flowing from the channel to the MTJ FM layer, S_{out}=S_{in}; on the contrary, if I_{inj} is positive, S_{out}= not (S_{in}).

Based on this principle, we can now considered more complex ASL circuits composed of multiple inputs connected through channels and channel-crossing. Figure 2.a represents a reconfigurable 3-input gate: I_{n1} and I_{n2} MTJs are input data and MTJ *F* is the cell configuration. At least two inputs have the same polarity and this circuit can be sized to guaranty that the spin current in the detector I_{sdet}, will switch the *Int* MTJ state with a current larger than the critical current depending on the state of this two identical inputs. Consequently, this gate implements the majority function, i.e. *Out2* =Maj(*In1*, *In2*, *F*). It can be shown that, with one input considered as a configuration function, and depending on the injected current polarity, all binary functions are available except XOR and XNOR.

XOR and XNOR function can be configured with majority function *Out2*=Maj(*In1*, *In2*, *F*, *Int'*, *Int'*), with *In1*/*In2*/*F* as three inputs, *Out1* and *Out2* as the outputs and *Int* as the intermediate MTJ. For the implementation of this majority function, we rely on the channel connecting MTJ Int to MTJ *Out2*. It is then a 2 steps computation: first, currents are injected into the three input terminals to switch both *Int* and *Out2* state ; then, a current is injected into *Int* (while the current is maintained in the three input MTJs), which switches *Out1* and also *Out2* state. Finally, a voltage is applied onto the *Out1* and *Out2* to detect the spin state. Table I summarizes the functions that can be implemented on *Out1* and *Out2*, according to input configurations *F*.

The design of such a cell is tedious since it requires to specify MTJ injection currents, taking into account numerous parameters such as the length of the channels and the used materials. Furthemore, it is required to take into account the placement of the MTJs in the layout. For instance, Figure 2.b illustrates an alternative layout, for which data input and output MTJs are aligned. Such a layout faciliates cells cascading, as illustrated in Figure 3.

TABLE I. RECONFIGURABLE CELL FUNCTIONS SET

Function *Out1*	Function *Out2*	Configurations	
		F	I_{inj}
$\overline{In1 \cdot In2}$	*In1*⊕*In2*	0	N
In1. *In2*	*In1*⊙*In2*	0	P
$\overline{In1 + In2}$	*In1*⊙*In2*	1	N
In1+*In2*	*In1*⊕*In2*	1	P
Min(*In1*,*In2*,*F*)	XOR3	#	N
Maj(*In1*,*In2*,*F*)	XNOR3	#	P

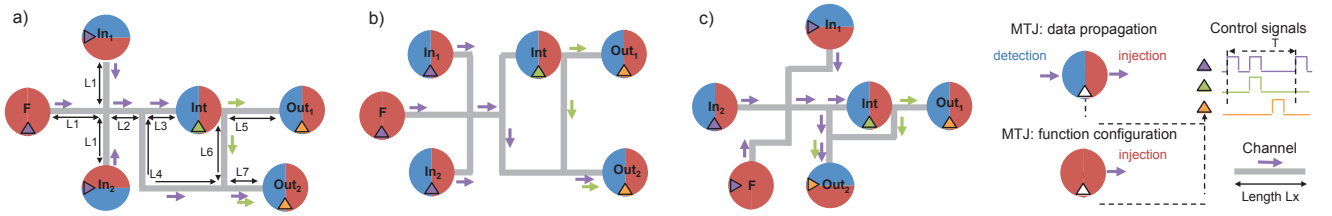

Fig. 2. ASL reconfigurable 3-input gate.

In the figure, channels allow connecting the outputs of the first cells to the inputs of the second cells, which allows using the same control signals to synchronize the computations in the cells. However, it is costly since signals outputted by the first cells need to be latched before computation in the second cells. For this purpose, we consider the cascading strategy illustrated in Figure 3.b, which leads to i) MTJs sharing between adjacent cells and ii) a shift of the control signals. The following section further discusses the impact of a cell's layout on the interconnect topology.

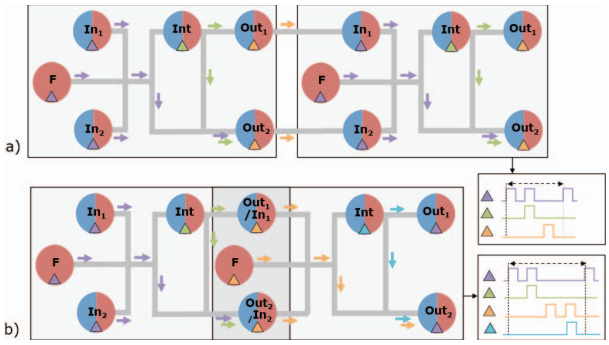

Fig. 3. ASL gate cascading strategies and clocking scheme.

III. System Design

While ASL cells with partial function set are interesting for MTJ count saving, it is also important to investigate the design and the programmability of larger scale architectures involving many-cells. A key factor for a realistic architecture is to keep the design as regular as possible, which calls for the replication of a same reconfigurable cell. Island-style architecture, which has been used in FPGA for decades, relies on a reconfigurable interconnect. While such organization leads to a very flexible solution by allowing to reach most of the cells, the power and area overhead induced by the interconnect leads to a poor energy efficiency and to a low computing density. Scalable and regular interconnects for nano-grain cells were introduced in [9] with the cell-matrix architecture. The authors proposed an integrated architecture combining both computational and control parts at matrix level offering a better scalability than FPGAs. More recently, static matrices have been considered in the context of nanograin architectures based on double gate FET [10]. While matrices interconnects are static and lead to a lower programming efficiency compared to island-style (i.e. not all the cells are reachable), their lower area and power overheads allow for massive scale integration of cells, which in turns improves the computing density. Matrices are thus the interconnect solution we consider. In the following, we first investigate matrices topologies and discuss their pros and cons. We then introduce challenges related to pipelining execution and to the matrices programming.

We investigate the use of existing topologies to statically interconnect ASL-cells. A major constraint is related to channel crossings, which is so far widely limited by the technology since spins transmitted on a channel diffuse on crossed channels: data thus cannot be transmitted independently from each other if the interconnects involve channel crossings. In order to tackle this limitation, we propose to build matrices relying on i) available data transmission inside the cells and ii) the location of the MTJ interfaces in the cells. The main goal is thus to assembly cells by using output MTJs of a cell as input MTJs of others cells. Figure 4.a illustrates a matrice for which cells assembly leads to a butterfly topology (we do not show intermediate and function configuration MTJ for sack of clarity). In this topology, data transmit unidirectionally from West to East, which calls for input and output MTJs placed on opposite sides of a cell. Since MTJs are not shared between cells located on a same column, there is no vertical data propagation, which could be a limitation for the programming step. For this purpose, diagonal data propagation is achieved by shifting the next column. This leads to matrice intermediate inputs and outputs, which could be useful to extract partial results or to inject data computed by another matrices. The configuration of the cells is static: it does not evolve during the execution of the application, i.e. the MTJs used for function configuration permently inject a spin current in the channel. As previously detailed, spin detection (toward an output MTJ) can operate only if spin injection (from an input MTJ) is maintained during the process. Since the output MTJs of a stage are the input MTJs of the following stage, odd and even column must operate under a phase shifted clock signals.

Figure 4.b illustrates a systolic topology-based matrice, which is designed following the same strategy. In this example, the 4 inputs/outputs are located on each size of the cell, which allows propagating data from North to South and from West to East. This topology implies diagonal clock domains. Figure 4.c illustrates a topology derived from systolic, in which cells located in odd columns are rorated. This allows propagating data from South to North. Comparing the topologies is not obvious since, at system level, the achieved computing density depends on application mapping efficiency and, at device level, the cells imply different layouts, as detailed in the following.

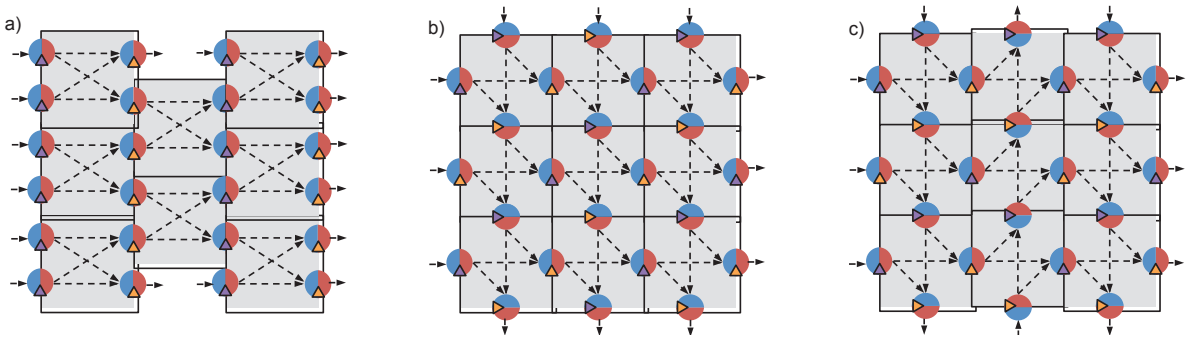

Fig. 4. Matrix topologies for ASL array.

At system level, numerous parameters can be explored and include, among other, the number of cell's inputs/outputs, the cell's available operators, the width and the length of the matrix. Design space exploration needs to be carried out and a generic programming tool is needed to configure such matrices. For instance, MPACK [11] is an additional module of VTR allowing to map any functions on nano-grain cells interconnected with a multi-stage interconnect topology. It allows partitionning an application into available operators and mapping the operatots on the matrices, taking into account the data dependencies and the interconnect topology. Similar tool considering ASL-specific properties is thus needed. In particular, such tool should handle multi-clock domains required by ASL, as detailed in the following.

At device level, a major challenge for the design of such a matrice is related to the cell's layout, which is strongly driven by the intended topology. For butterfly topology, the input MTJs of the first column must be aligned and consecutive MTJs but be equidistant from each other. The same constraint is applied to the output MTJs of the first column, i.e. the input MTJs of the second column. A possible layout has been illustrated in Figure 2.b.Systolic topology leads to even more layout constraints due to the 2D mesh MTJ organization it involves. The cell used for Systolic topology is illustrated in Figure 2.c. Designing cells under such layout constraints calls for the modular modeling approach previsouly described: starting from the location of the interfaces, parameters such as channel length, channel length ratio and injection current are explored to design the cells. Such a circuit has not yet been experimentally demonstrated. It is therefore necessary to provide some evidence of its feasibility by simulating a sufficiently faithful and convincing model.

IV. MODULAR MODELLING

Table II shows several ASL compact modeling. These ASL modeling is based on the spin-circuit concept, which is proposed in [12]. In this model, each device is represented as a pi-network that contributes to the conductance matrices, which allows ASL-based circuits implementation and analysis. However, such approach is not scalable since complex circuits design leads to large-scale matrices that need to be carefully established. Furthermore, the model is implemented in MatLab, which is not a suitable platform for the circuit designer community. In [13], the authors proposed a compact model integrating all the devices into a single block using a set of predefined equations. Such model is not suitable for hierarchical circuit design since the bloc is specific to a given cell. A circuit simulation environment relying on electrical circuit elements has been proposed [14] and an ASL model has been implemented with Verilog-A [15]. Both approaches enable the design and the simulation of spintronic devices-based circuits using a circuit solver such as HSPICE. However, they do not integrate important characteristics such as spin diffusion delay and channel breakdown effects. Moreover, these modeling relies on the non-collinear magnetoelectronic theory [16]. It disregards the interface spin-flip scattering and uses finite-element formulation which is implicit for the definition of current-voltage definition.

Different from existing models, we propose a comprehensive compact model that relies on the generalized Maxwell's equations adapted to the spin domain [8]. This leads to an explicit definition of current and voltage relations in circuits, which thus ensure generic design solution. This very flexible model has been decomposed into blocks corresponding to injector/detector, channel, ground and contact devices. The spin transfer torque (STT) effect: switching threshold current and dynamic switching delay, is integrated into the injector and detector, with the expressions in [17]. Moreover, our ASL device contains MTJs as injector/detector and non-volatile memories, the Tunnel MagnetoResistance (TMR) effect is also integrated in the injector/detector model. Furthermore, our model takes into account channel breakdown and channel spin diffusion delay effects. This allows preventing from destructive design options and helps the designer calibrating the devices (e.g. injection current specification). Our model is implemented with Verilog-A on Cadence platform, which allows designing hierarchical and complex circuits in an environment already used by the designer community.

V. SIMULATION RESULTS

Based on the developed compact model, we use Cadence to run transient simulations for the architecture in Figure 2, We consider 40 nm MTJs and 10 µm for the channel spin diffusion length. We assume the channel length in Table III, considering the dipolar coupling. For the injection current, we first set the I_{inj} to the three inputs to 900 µA, which leads to the maximum spin current allowed in the channel.

TABLE II. ASL COMPACT MODELING COMPARISON

Integrated effect	**Srinivasan[12]**	**Bonhomme[14]**	**Calayir[15]**	**Su[13,18]**	**Our model**
Language/software	MatLab	Circuit simulator	Verilog-A	Verilog-A/Cadence	Verilog-A/Cadence
Scalability	Low	Low	High	×	High
Module	Separated conductance matrix	Separated electrical circuit element	Separated conductance block	One block	Separated conductance block
STT effect	Coupled with LLG equation (in-plan)	Coupled with LLG equation	Coupled with LLG equation (perpendicular)	Integrated the static and dynamic switching expressions (perpendicular)	Integrated the static and dynamic switching expressions (perpendicular)
TMR effect	×	×	×	√	√
Channel breakdown effect	×	×	×	√	√
Channel diffusion time	×	×	×	×	√
Spin relaxation in FM	×	×	√	×	√
Fundamental theory	[16]	[16]	[16]	[16]	[8]

Then, considering the length of the channels, the weights needed to compute *Out2* (the weight of *Int* is twice the weight of *In1, In2* and *F*) and the current division from *Int* to *Out1* and *Out2*, I_{injInt} to *Int* is set to 1.44 mA. We assume the ferromagnetic damping factor α=0.007 relies on a conservative technology. As shown in Figure 5, the functional behavior of this architecture is validated, with V_{write} as input terminal state writing, I_{inj} as injection current into the three input terminals, I_{injInt} as current injected into *Int* terminal, V_{read} as the reading voltage, *In1*/*In2*/*F* as input states, *Int*/*Out1*/*Out2* as output states.

TABLE III. SIMULATION PARAMETERS

Parameter	**Value**
L1	50 nm
L2	50 nm
L3	50 nm
L4	80 nm
L5	100 nm
L6	50 nm
L7	50 nm
Iinj	900 µA
IinjInt	1.44 mA

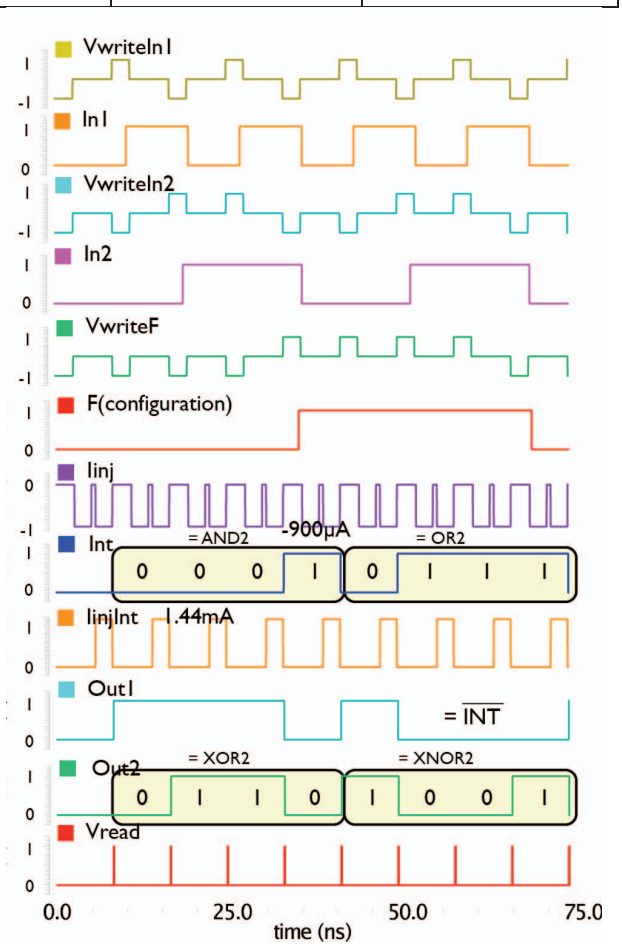

Fig. 5. Simulation of the reconfigurable gate. AND2(XOR2) and OR2(XNOR2) functions are simulated with configurations F=0 and F=1 respectively.

VI. Conclusion

Thinking about ASL as a major spintronics milestone requires taking its specific features into account in the circuit and system design strategies. Copying CMOS circuits is inefficient, as this method deals poorly with spin current fanout, which vanishes in a network containing forks and long interconnects. In this paper, we present a high-density array architecture of programmable ASL abutted cells to minimize interconnect overhead. Using an efficient modular modelling approach that computes both spin and charge currents concurrently, the programmable cell design is presented and simulated. In addition, because the Boolean functions available rely intrinsically on the majority function, the design tools must also be reworked in accordance with the specific constraints associated with spintronic devices. This constitutes the next steps of our work.

Acknowledgment

This work was supported by China Scholarship Council (CSC) program.

References

[1] Shannon, C. E. (1938). "A Symbolic Analysis of Relay and Switching Circuits". Trans. AIEE. 57 (12): 713–723..

[2] S. Wolf, D. Awschalom, R. Buhrman, J. Daughton, S. von Molnar, M. Roukes, A. Chtchelkanova, and D. Treger, "Spintronics: A spin-based electronics vision for the future," Science, vol. 294, pp. 1488–1495, 2001.

[3] C. Chappert, A. Fert, F. Nguyen Van Dau "The emergence of spin electronics in data storage.," vol. 6, no. 11, pp. 813–823, Nov. 2007.

[4] S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H. D. Gan, M. Endo, S. Kanai, J. Hayakawa, F. Matsukura, and H. Ohno, "A perpendicular-anisotropy CoFeB-MgO magnetic tunnel junction.," Nat. Mater., vol. 9, no. 9, pp. 721–724, 2010.

[5] Y. Zhang, W. Zhao, Y. Lakys, J. Klein, J. Kim, D. Ravelosona, and C. Chappert, "Compact modeling of perpendicular-anisotropy CoFeB/MgO mangetic tunnel junctions," IEEE Trans. Electron Devices, vol. 59, no. 3, pp. 819–826, 2012.

[6] E. I. Rashba, "Theory of electrical spin injection: Tunnel contacts as a solution of the conductivity mismatch problem," Phys. Rev. B - Condens. Matter Mater. Phys., vol. 62, no. 24, pp. 2–5, 2000.

[7] G. Schmidt, L. W. Molenkamp, a. T. Filip, and B. J. van Wees, "fundamental obstacle for electrical spin-injection from a ferromagnetic metal into a diffusive semiconductor," Phys. Rev. B, vol. 62, no. 8, p. 4, 1999.

[8] J. Fabian and I. Zutic, "The standard model of spin injection," *arXiv preprint arXiv:0903.2500*, 2009.

[9] Lisa J K Durbeck and Nicholas J Macias. The Cell Matrix: architecture for nanocomputing. Journal of Nanotechnology, Volume 12, Issue number 3, starting page 217, 2001.

[10] Nataliya Yakymets, Ian O'Connor, Kotb Jabeur and Sebastien Le Beux. Multi-Level Mapping of Nanocomputer Architectures Based on Hardware Reuse. IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS), special issue on Computing in Emerging Technologies, 2014

[11] Pierre-Emmanuel Gaillardon, Haykel Ben-Jamma, Fabien Clermidy, Ian O'Connor. Ultra-fine-grain FPGAs: A granularity study. Proceeding of the International Symposium on IEEE/ACM NANO-Arch, page 9 to 15, San Diego, 2011

[12] S. Srinivasan, V. Diep, B. Behin-Aein, A. Sarkar, S. Datta, D. Awschalom, J. Nitta, and Y. Xu, "Modeling multi-magnet networks interacting via spin currents," arXiv preprint arXiv:1304.0742, 2013.

[13] L. Su, W. Zhao, Y. Zhang, D. Querlioz, Y. Zhang, J.-O. Klein, P. Dollfus, and A. Bournel, "Proposal for a graphene-based all-spin logic gate," Applied Physics Letters, vol. 106, no. 7, p. 072407, 2015

[14] P. Bonhomme, S. Manipatruni, R. M. Iraei, S. Rakheja, S.-C. Chang, D. E. Nikonov, I. A. Young, and A. Naeemi, "Circuit simulation of magnetization dynamics and spin transport," IEEE Transactions on Electron Devices, vol. 61, no. 5, pp. 1553–1560, 2014.

[15] V. Calayir, D. E. Nikonov, S. Manipatruni, and I. A. Young, "Static and clocked spintronic circuit design and simulation with performance analysis relative to CMOS," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 61, no. 2, pp. 393–406, 2014.

[16] A. Brataas, G. E. Bauer, and P. J. Kelly, "Non-collinear magnetoelectronics," Physics Reports, vol. 427, no. 4, pp. 157–255, 2006

[17] Y. Zhang, W. Zhao, Y. Lakys, J.-O. Klein, J.-V. Kim, D. Ravelosona, and C. Chappert, "Compact modeling of perpendicular-anisotropy CoFeB/MgO magnetic tunnel junctions," IEEE Transactions on Electron Devices, vol. 59, no. 3, pp. 819–826, 2012.

[18] L. Su, Y. Zhang, J.-O. Klein, Y. Zhang, A. Bournel, A. Fert, and W. Zhao, "Current-limiting challenges for all-spin logic devices," Scientific reports, vol. 5, p. 14905, 2015.