ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Electronic Engineering (33.th cycle)

# Exploring logic-in-memory architectures with skyrmion technology

**Luca Gnoli**

* * * * *

**Supervisor**
Prof.ssa Mariagrazia Graziano

**Doctoral Examination Committee:**
Prof. Liliana Buda-Prejbeanu, Univ. Grenoble Alpes,
Prof. Gyorgy Csaba, Pazmany Peter Catholic University,
Prof. Massimo Ruo Roch, Politecnico di Torino,
Prof. Markus Becherer, Technical University of Munich,
Prof. Gianluca Piccinini, Politecnico di Torino

Politecnico di Torino
September , 2021

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Luca Gnoli
Turin, September , 2021

# Summary

From their experimental verification in 2009 magnetic skyrmions attracted great interest thanks to many intriguing characteristics that this magnetic texture has. Their small dimension suggested a possible application in high density data storage. Their stability and the chance to be moved with low current densities compared with other textures suggested their use as successors of domain walls in racetrack memories. In addition their rich dynamics linked to current stimulus, opened the door to many other applications, starting from boolean logic circuits up to non-boolean computation devices. The vast majority of research on applications of skyrmion technology focused its attention on the proposal of single devices, studying basic logic functions to demonstrate the feasibility of logic with skyrmion technology. At the same time, the study of complex systems based on this technology is still lacking, only few examples in literature have been presented. For other technologies, both electric and magnetic, an important step in their evolution has been the study of the challenges and possibilities that arise from the implementation of logic systems able to solve complex problems. The design of a complex system, indeed, permits to better appreciate the strength of a technology but also to reveal possible drawbacks linked to circuit integration. Moreover, the realization of these systems is useful on one side to have a comparison with already tested and known technologies and on the other to shed light on the design challenges that might appear implementing complex circuits in a specific technology. The presented thesis has the goal to address this issue for skyrmion technology, providing a wider view on skyrmion systems. The thesis starts from the study of possible strategies for complex circuit design, based on skyrmions and later proposes designs in skyrmion technology with increasing complexity. In particular, the design proposals, focuses on logic in memory architectures. Skyrmions are indeed a promising memory technology. This aspect, together with the possibility to realize logic functions, suggests a preferable application in architectures that embeds both non-volatile memory and logic. Finally, another way to design magnetic circuits employing skyrmion technology has been explored. The proposal done in the last chapter is based on a circuit structure in which different magnetic technologies can successfully communicate and realize different tasks. This connection allows to design circuits in which different operations of the algorithm are accomplished by different technologies, each one optimized for its own application. The proposal in particular focuses on the demonstration of the interface used to put in communication Perpendicular Nanomagnetic Logic and Skyrmion technology. In the proposed interface the skyrmion technology, that is the memory technology, can pass the information stored in form of skyrmion to perpendicular nanomagnetic logic that realize the logic part of the circuit. The opposite conversion is also demonstrated: the information coming from the Perpendicular Nanomagnetic Logic is translated in skyrmion form to be stored back in memory. This direct interface between the two technologies opens interesting scenarios in which magnetic logic can potentially access with very high bandwidth the magnetic storage without any electrical conversion needed.

# Acknowledgements

I would like to acknowledge professor Mariagrazia Graziano for giving me the possibility to work on this thesis and for the guidance and support during these years.

Then I would like to acknowledge professor Marco Vacca for the scientific stimulus, the fruitful discussions and the support he gave me starting from the master thesis to the present days.

I would like then to acknowledge PhD. Fabrizio Riente for the scientific support, great collaboration and push in the research he gave me especially in the last years.

I would like to acknowledge Prof. Markus Becherer, Valentin Ahrens, Simon Mendisch and Martina Kiechle for the collaboration and the fruitful discussions.

Finally I would like to acknowledge the members of the VLSI Lab group for the great environment that they created and maintained during these years.

*I would like to dedicate this thesis to my family and to Ilaria*

# Contents

List of Tables XI

List of Figures XII

1 Introduction 1
1.1 Micromagnetism . . . 1
1.1.1 LLG equation . . . 1
1.1.2 Magnetic energies . . . 2
1.1.3 Exchange Energy . . . 3
1.1.4 Demagnetizing energy . . . 4
1.1.5 Anisotropy Energy . . . 4
1.1.6 Zeeman Energy . . . 7
1.1.7 Thermal Energy . . . 7
1.2 Skyrmions . . . 8
1.2.1 Magnetic skyrmions types . . . 8
1.2.2 Topological charge . . . 9
1.2.3 Vorticity number . . . 9
1.2.4 Helicity number and polarity . . . 10
1.2.5 Nucleation . . . 11
1.2.6 Detection . . . 12
1.2.7 Skyrmion dynamics . . . 13

2 Design flow for complex circuits based on skyrmions 17
2.1 Motivation . . . 17
2.2 Skyrmion devices . . . 20
2.2.1 Skyrmion Domain-Wall logic . . . 20
2.2.2 Skyrmion gates based on skyrmion-skyrmion and skyrmion-edge interaction . . . 23
2.2.3 Skylogic . . . 24
2.2.4 Skyrmion gates synchronization . . . 26
2.3 Modeling of skyrmion devices in VHDL . . . 27
2.3.1 Basic library . . . 28

2.3.2 Gate basic structure . . . 28
2.3.3 Toy model . . . 30
2.3.4 Skyrmion conservative logic simulation example . . . 32
2.3.5 Design flow . . . 33
2.3.6 Conclusion and future prospects . . . 36

**I Logic in memory devices with skyrmions 37**

**3 Content addressable memories with skyrmions** 41
3.1 Content Addressable Memories . . . 41
3.2 CAM devices with skyrmion technology . . . 42
3.2.1 CAM Cell design . . . 43
3.2.2 Skyrmion CAM . . . 44
3.2.3 Input bit restore . . . 46
3.2.4 Methods . . . 47
3.2.5 Integration in memories . . . 48
3.2.6 Micromagnetic simulations and performance evaluation . . . 50
3.3 Partial conclusions and improvements . . . 54

**4 Final conclusions and prospects** 55

**5 Minimum Maximum Search** 57
5.1 Algorithm . . . 57
5.2 Memory Cell . . . 58
5.3 Cell operation . . . 60
5.4 Memory organization . . . 61
5.5 Logic-in-memory . . . 62
5.5.1 Maximum/Minimum Search Operation . . . 64
5.5.2 Additional logic operations . . . 65
5.6 Methods . . . 66
5.7 Results . . . 67
5.8 Final considerations . . . 70

**6 Advanced Encryption Standard** 71
6.1 AES algorithm . . . 72
6.1.1 Key expansion . . . 72
6.1.2 Add Round Key . . . 75
6.1.3 Sub Bytes . . . 75
6.1.4 Shift Row . . . 76
6.1.5 Mix Columns . . . 76
6.2 AES logic-in-memory with skyrmions . . . 78
6.2.1 LIM device structure . . . 78

6.2.2 Array Design . . . 79
6.2.3 Memory Cells interconnect . . . 80
6.2.4 Memory Cell Design . . . 83
6.2.5 Control Logic . . . 86
6.2.6 Current generation . . . 93
6.2.7 Integration with skyrmion memories . . . 94
6.3 Simulations and performance evaluation . . . 95
6.3.1 Methods . . . 95
6.3.2 Performance evaluation . . . 95
6.3.3 Final considerations . . . 99

**7 Skyrmion LIM: Final considerations and future works** 101

**II PNML-Skyrmion Interface** **105**

**8 Skyrmion-PNML interface** 107
8.1 Motivation . . . 107
8.2 Perpendicular Nanomagnetic Logic . . . 108
8.2.1 Nanomagnetic Logic . . . 108
8.2.2 Perpendicular Nanomagnetic Logic . . . 109
8.3 Skyrmion-PNML device . . . 112
8.3.1 Information stability . . . 113
8.3.2 Skyrmion to PNML conversion . . . 117
8.3.3 PNML to Skyrmion conversion . . . 118
8.3.4 Methods . . . 118
8.3.5 Micromagnetic simulations . . . 119
8.3.6 Energetic evaluations . . . 122
8.4 Final considerations and future works . . . 123

**9 Final conclusions and prospects** 125

**Bibliography** 136

# List of Tables

2.1 Computation time required by mumax3 on GPU for 5 ns of simulation 18
2.2 Parameters used for micromagnetic simulations used as base for modeling . . . 32
3.1 Parameters used for micromagnetic simulations and current distribution computation . . . 44
3.2 Performance comparison between skyrmion based CAM and CMOS implementation - 32 x 32 CAM array . . . 53
3.3 Performance comparison between skyrmion based CAM and other non-volatile CAM implementations . . . 53
5.1 Parameters used for micromagnetic simulations and current distribution computation . . . 66
5.2 AREA and TIMING, WORD WIDTH = 64 bit . . . 67
5.3 Power consumption of a single memory cell for each phase . . . 68
5.4 POWER AND ENERGY - WORD WIDTH=64 bit . . . 70
6.1 Parameters used for micromagnetic simulations and current distribution computation . . . 95
6.2 Comparison of AES LIM implementations . . . 97
8.1 Parameters used for micromagnetic simulations and current distribution computation . . . 114
8.2 Parameters used for micromagnetic simulations and current distribution computation . . . 119
8.3 Energy performance of Skyrmion-PNML conversion . . . 122

# List of Figures

1.1 Evolution of the magnetic moment direction in response to an effective field. . . . 2

1.2 Dzyaloshinskii–Moriya interaction in bulk materials (a) and at the interface between a ferromagnet and an high spin orbit metal (b) . 4

1.3 Energy distribution as function of the magnetization direction a) Exchange interaction normalized energy distribution b) Uniaxial anisotropy energy distribution with positive first order anisotropy constant resulting in an easy axis anisotropy c) Uniaxial anisotropy energy distribution with negative first order anisotropy constant resulting in an easy-plane anisotropy . . . . 5

1.4 Skyrmion textures. a) Bloch type skyrmion; b) Néel type skyrmion 9

1.5 Vorticity computation for different magnetic configurations. . . . . . 10

1.6 Nucleation of skyrmion by means of vertical polarized current in a circular dot of CoFeB $\alpha = 0.015$, $D = 3\,\mathrm{mJ\,m^{-2}}$. Image reproduced from [31] . . . . 11

2.1 Design flow example using exclusively micromagnetic simulation for the datapath design . . . . 19

2.2 Skyrmion domain wall conversion. The figures have been extracted from [33] shows the conversion from domain to skyrmion (a-c,g-i) and the opposite process (d-f,j-l). The effect is guided by the Spint Transfer Torque (STT) effect. . . . . 21

2.3 Skyrmion domain wall logic. a) AND gate b) OR gate c) Duplication mechanism based on domain wall. Images extracted from [56] . . . 22

2.4 Skyrmion logic system based on skyrmion-edge and skyrmion-skyrmion repulsion effects. Images extracted from [55] . . . . 23

2.5 Reconfigurable skyrmion logic gate. The control for reconfigurability is realized by the two VCMA gates guided by $V_{K1}$ and $V_{K2}$ and by the $V_M$ value controlling the output read head. Image extracted from [57]. . . . . 25

2.6 Skylogic gates: a) NOT logic gate b) NOR logic gate. The $V_{in}$ voltage signals produce produce the input for the gate. The $V_{propx,y}$ signals control the movement of the skyrmion. At the output the $V_{out}$ signal is the output of the gate. Image extracted from [58] . . . . . . . . 25
2.7 Synchronization methods for skyrmions. A notch in the track pins the skyrmion. A pulse unpins the skyrmion **(a-b)** A VCMA zone stops the motion of the skyrmion. When voltage is removed the skyrmion can proceed in the VCMA zone. **(c-d)** . . . . . . . . . . . 26
2.8 Basic structures for library. a) Line; b) AND/OR gate; c) NOT/-COPY gate; d) JOIN gate; e) T structure; f) VCMA sync gate . . 29
2.9 Skyrmion gates interface . . . . . . . . . . . . . . . . . . . . . . . . . 29
2.10 Gate partition for confinement effect approximation. . . . . . . . . . 30
2.11 Logical structure of the modeled gate. . . . . . . . . . . . . . . . . . 31
2.12 Mapping of XOR based on skyrmion logic with the skyrmion library 33
2.13 Example of dynamical effect linked to skyrmion motion in systems with low damping. In a) the motion of the skyrmion in the track, in b) the evolution of the position of the center in time. The example is referred to a W/CoFeB/MgO with a damping constant of 0.015 and a current of $6 \times 10^{10}\,\mathrm{A\,m^{-2}}$. Due to the reduced damping, the skyrmion changes direction after the collision with the VCMA gates and bounces back. . . . . . . . . . . . . . . . . . . . . . . . . . . 35
3.1 CAM memory basic schematic . . . . . . . . . . . . . . . . . . . . . . 42
3.2 Reference stack for skyrmion movement. The curb confine the movement of the skyrmion to enhance its stability at high currents . . . 43
3.3 Skyrmion CAM Cell. In a) the cell with forward restore path to memory. In b) the restore path is in the memory direction . . . . . 44
3.4 Example of CAM Cell operation. In the example data input is 1 and SL is 1. (a)The operation is triggered. (b)The skyrmions continue their movement based on the mutual repulsion. (c)The search signal SL is equal to '1', the skyrmion is allowed to cross the VCMA gate. (d)The result is ready in the evaluation region. The result can be deleted or sent to memory . . . . . . . . . . . . . . . . . . . . . . . 45
3.5 Restore procedure for the CAM cell with forward restore (a-b) and the CAM cell with backward restore mechanism (c-d) . . . . . . . . 48
3.6 Possible connection schemes for CAM cells to the main memory. a) CAM cell connections on a single side. In this case, the restore mechanism sends the information on another skyrmion memory track. b) CAM cell connections on both sides. . . . . . . . . . . . . . . . . . . 49
3.7 Micromagnetic simulations of the skyrmion CAM cell. (a)Stored bit=0 SL=0. (b)Stored bit=0 SL=1. (c)Stored bit=1 SL=0. (d)Stored bit=1 SL=1. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50

3.8 a)Skyrmion CAM search time and b)power and energy performance as functions of the driving current density. · . . . . . . . . . . . . . . 51
5.1 In a), the contacts for current generation in the memory cell. The grey area indicates the tungsten layer. For every operation a different set of contacts is involved to generate the required local currents to move the skyrmion. In b), the path followed by a skyrmion entering in the processing zone in case a 1 is stored in the cell and the mask bit is set to 1. Image extracted from [85] . . . . . . . . . . . . . . . . 60
5.2 In figure the four phases of elaboration of the stored skyrmions. In a) the information is moved from the memory track into the duplication gate. In b) the information is duplicated by means of Skyrmion-DW conversion. In c) the information is processed on top and moved back to the memory cell on bottom. In d) the result of the operation is read on the left, the information is restored into the cell on the right. Image extracted from [85] . . . . . . . . . . . . . . . . . . . . 61
5.3 Read and write mechanism. The track has more than one read head to reduce the read operation latency. In order to sense the skyrmion presence under the read-head, the value produced is compared with a reference Image extracted from [85] . . . . . . . . . . . . . . . . . . 62
5.4 logic-in-memory architecture schematic. On the left the block diagram of the complete circuit. On the right the basic cell of the array. The skyrmion is stored inside the memory in correspondence of the red dot.Image extracted from [85] . . . . . . . . . . . . . . . . . . . . 63
5.5 FSM State diagram for Maximum/Minimum search Image extracted from [85] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
5.6 Logical operation between rows and columns. a) Column-wise operation b) Row-wise operation. [85] . . . . . . . . . . . . . . . . . . . . 65
5.7 Delay area product with respect to the memory size of the proposed skyrmion implementation and CMOS [84]. Image extracted from [85] 68
5.8 Current density produced during duplication operation. The current path highlighted in green is the one required to move the skyrmion. The current path in red indicates the additional path for the current not needed for the current. This additional paths increase the power consumption in the operation. Image extracted from [85] . . . . . . 69
6.1 AES input state matrix representation . . . . . . . . . . . . . . . . . 72
6.2 AES encryption algorithm general block diagram . . . . . . . . . . . 73
6.3 Key expansion algorithm procedure. The example shown is referred to a 128-bit key producing 44 words from the cipher key . . . . . . 74
6.4 Add round key operation. RK matrix refers to the round key computed by the key expansion operation. . . . . . . . . . . . . . . . . . 75

6.5 Sub Bytes operation. The input word is substituted with another word using half of the word as row index and half of the word as column index . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76
6.6 Mix column operation . . . . . . . . . . . . . . . . . . . . . . . . . 77
6.7 Block diagram of the complete LIM device. . . . . . . . . . . . . . . 79
6.8 Word organization in the AES LIM array with respect to the reference state matrix. a) Reference state matrix b) LIM array mapping c) Bit position inside a single word . . . . . . . . . . . . . . . . . . 81
6.9 Interconnection scheme of two neighbor words in the LIM array. (a) Crossing matrix used for AES LIM implementation. The yellow zones corresponds to VCMA gates used to control skyrmion direction 82
6.10 AES memory cell. The highlighted squares are VCMA gates . . . . 83
6.11 XOR logic implemented in the LIM Cell. The information for the A and B input of the input can be produced from different sources. The white box indicate external write heads for input generation . . 84
6.12 Insertion of information in auxiliary storage position. The auxiliary position is used as temporary storage for partial computations. . . 85
6.13 Master FSM states diagram. The FSM manage AES, write and read operations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
6.14 Add round key and sub-bytes FSM states diagram. The FSM manage AES, write and read operations . . . . . . . . . . . . . . . . . . 87
6.15 Shift row FSM states diagram. . . . . . . . . . . . . . . . . . . . . . 88
6.16 ShiftRow operation example. The rotation moves the word 1 in position of word 4 and shifts the other 3 words. The example reports 3 bits words, the algorithm is executed on 8 bits word. . . . . . . . 89
6.17 MixColumns FSM state diagram . . . . . . . . . . . . . . . . . . . . 90
6.18 Partial products operation example. The target cell in this example is the cell 4. The row indication refers to the correspondent position in the reference state matrix. . . . . . . . . . . . . . . . . . . . . . . 92
6.19 Contact scheme for the current generation of the AES LIM device . 93
6.20 Vertical and horizontal topology for LIM device integration with skyrmion racetrack memory . . . . . . . . . . . . . . . . . . . . . . 94
6.21 Critical path through the basic LIM cell . . . . . . . . . . . . . . . 96
7.1 Crossing matrix used for AES LIM implementation. The yellow zones corresponds to VCMA gates used to control skyrmion direction 103
8.2 3-Inputs PNML majority voter. . . . . . . . . . . . . . . . . . . . . . 111
8.3 2-Inputs PNML logic gate. The function realized by the logic gate is defined by an external bias field in positive or negative z-direction. 111
8.4 Skyrmion-PNML device. . . . . . . . . . . . . . . . . . . . . . . . . 113
8.5 Skyrmion radius evolution with an applied external magnetic field. The skyrmion is confined in a plane with dimensions 256 nm × 256 nm 114

8.6 Skyrmion confinement structures. **(a)** plain racetrack **(b)** curbed racetrack . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 115
8.7 Skyrmion radius evolution with a negative external field. The skyrmion is confined in a 40 nm strip and in 50 nm . . . . . . . . . . . . . . . 116
8.8 Skyrmion radius evolution with an applied external magnetic field. The skyrmion is confined in a curbed track wide 30 nm and deep 0.5 nm . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 116
8.9 Skyrmion-PNML conversion structure. The yellow structure on top is the input for the PNML logic . . . . . . . . . . . . . . . . . . . . . 117
8.10 PNML-Skyrmion conversion structure. The yellow structure on top is the the PNML logic . . . . . . . . . . . . . . . . . . . . . . . . . . 118
8.11 Skyrmion to PNML input conversion. The skyrmion in proximity the constriction (a) is pushed by means of a current and is converted in a domain(b). The domain is the expanded by means of a magnetic field. A nucleation is triggered in the top structure. A domain in the top structure is nucleated (c). The domain expands in the top structure (d). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120
8.12 PNML to skyrmion conversion conversion. The PNML output is in the bottom direction and influence the ANC in the bottom structure(a). A domain is nucleated in the bottom structure with the same polarity as the top input (b). The domain expands is the output structure by means of a current. A positive field is applied in this phase (c). A new skyrmion domain detach from the constriction. The skyrmion is pushed away from an external current (d) . . . . . 121
9.1 Basic movement mechanism of magnetic bubbles. a) When the magnetic field at the end of the bar points in the opposite direction of the bubble core the object is attracted. b) When the magnetic field at the end of the bar points in the same direction of the bubble core the object is repelled. . . . . . . . . . . . . . . . . . . . . . . . . . . 130
9.2 Moving mechanism for bubble logic. Images reproduced from [135] 131
9.3 Magnetic bubble logic gates. a) The AND/OR gate produces at the bottom side the OR results and on the right the AND result between the two inputs comping from the top and left track. The operation is executed with the natural repulsion of bubbles. b) and c) Dynamic and static deflectors. Both the elements determine the path of the input bubble present on input a based on the bubble present in b. The information stored in b can be dynamically or statically placed at the b input. d) Majority gate based on chevrons stacks. Images reproduced from [136] . . . . . . . . . . . . . . . . . . . . . . . . . 132
9.4 Bubble logic based on minterms. Image reproduced from [137] . . . 133
9.5 Bubble associative search. Image reproduced from [140] . . . . . . . 134

9.6 A skyrmion NOT/COPY element A) and the basic dynamical deflector B). As can be seen the NOT/COPY gate is equivalent to a dynamical deflector for bubble logic . . . . . . . . . . . . . . . . . . . . 135

9.7 Classification of different logic elements in bubble technology. Image reproduced from [141] . . . . . . . . . . . . . . . . . . . . . . . . . 135

# Chapter 1

# Introduction

## 1.1 Micromagnetism

There are several descriptions that can be used to represent magnetic systems. Originating at quantum level and being described up to very large scales, magnetism can be described at different scales depending on the target of the study and on the grade of approximation acceptable. The presented study deals with static and dynamic characteristics of magnetic textures in ferromagnets. For this reason the most suitable approach is the micromagnetism. Micromagnetism is a theory regarding magnetization in magnetic materials formalized by Brown [1] that describes the evolution of magnetization considering constant the magnetic characteristics of the material over small volumes with dimensions higher than atomic distances. The magnetic contributions of atomic structures are averaged over this volume resulting in a single magnetization vector describing the complete volume. In this approach the magnetization is considered a continuously varying quantity. This kind of approximation allows to not consider explicitly the atomic structure of the material and simplify the calculation obtaining under some assumptions a good agreement with experimental results. With this description energy minimization problems can be solved allowing to study magnetic domains. At the same time, taking advantage of dynamical equations, the evolution of the system can be studied in time.

### 1.1.1 LLG equation

When considering the evolution of magnetic moments in time, an accurate description of the magnetization dynamic can be done with the Landau-Lifshitz-Gilbert (LLG) equation. The equation is the following [2, 3]

$$\frac{d\boldsymbol{m}}{dt} = -\mu_0\gamma(\boldsymbol{m} \times \boldsymbol{H_{eff}}) + \alpha(\boldsymbol{m} \times \frac{d\boldsymbol{m}}{dt}) \tag{1.1}$$

where $\boldsymbol{m}$ is the magnetization normalized by the saturation magnetization of the material, $\gamma$ is the electron gyromagnetic ratio, $\alpha$ is the damping parameter of the material and $H_{eff}$ is the effective magnetic field that embeds different contributions to the magnetization dynamics.

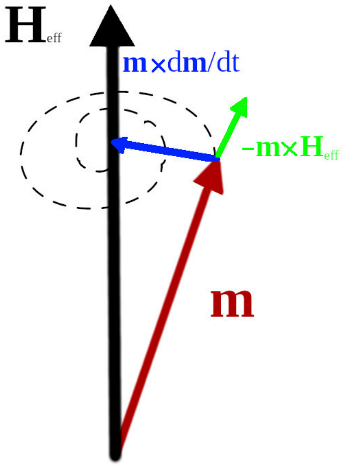


Figure 1.1: Evolution of the magnetic moment direction in response to an effective field.

In the equation the first term describes the precessional movement of magnetic moment around the direction of the effective field. The second term describes the relaxation of magnetization along the effective field direction. In the stable state the magnetization has the tendency to align with the effective field direction in order to minimize the magnetic energy of the system. The resultant movement of magnetization in response to an effective field is shown in figure 1.1 The effective field derives not only from external applied fields but from all the contribution to the energy of the magnetic moments. This field can be expressed as:

$$\boldsymbol{H_{eff}} = -\frac{1}{\mu_0 M_s}\frac{\partial E_V}{\partial \boldsymbol{m}} \tag{1.2}$$

where $E_V$ represents the Gibbs free energy energy density.

### 1.1.2 Magnetic energies

The allowed stable magnetic state in a ferromagnetic system depends on the energy associated with the required states. In addition the evolution of a particular system is strictly related to the energies acting on the system. Finally many information can be drawn from the energy landscape

### 1.1.3 Exchange Energy

Exchange energy originates from spin-spin interactions. The energy of interactions between neighbor spin $S_i$ and $S_j$ can be computed by means of the Heisenberg Hamiltonian as [4]

$$H = -2 * J * \hat{S}_i * \hat{S}_j \tag{1.3}$$

where $S_i$,$S_j$ are dimensionless spin operators and $J$ is the exchange constant b neighbor magnetic moments. From equation is possible to see how in case of positive exchange constants the lowest energy state is achievable for parallel spin direction. In this situation a ferromagnetic exchange between neighbor magnetic moments is present. In case the exchange constant J<0 the most favorable arrangement is for neighbor moments aligned in antiparallel direction. This exchange interaction is called antiferromagnetic exchange interaction. The energy density linked with exchange interaction is the following:

$$E_{ex} = A \cdot (\boldsymbol{\nabla m})^2 \tag{1.4}$$

where $A$ is the exchange stiffness of the material defined as $A \sim kJS^2/2$. In this definition, obtained considering only the closest neighbor and considering $S_i$ as a classical vector, $J$ is the nearest-neighbor exchange integral, $S$ is the spin magnitude, $a$ is the lattice constant, and $k$ is a factor that depends on the symmetry of the lattice and that is in the order of unity [4].

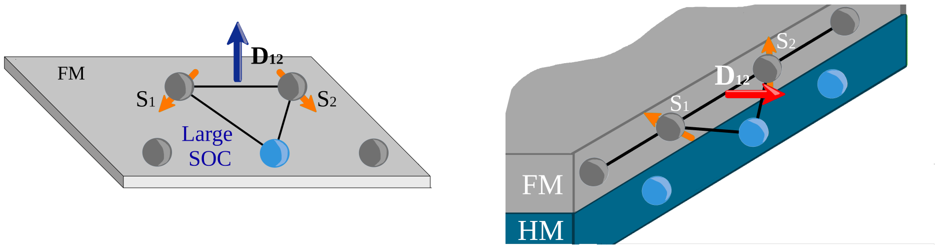
#### Dzyaloshinskii–Moriya interaction

In materials lacking inversion symmetry an additional exchange term is involved when computing the system energy: the Dzyaloshinskii–Moriya interaction (DMI). Given a system composed by two neighbor magnetic moments represented as $\boldsymbol{\hat{S}_i}$ and $\boldsymbol{\hat{S}_j}$ and a third element having a strong spin-orbit coupling interaction with both elements, as shown in figure 1.2 the Hamiltonian linked to this interaction is given by:

$$H_{DM} = \boldsymbol{D_{ij}} \cdot (\boldsymbol{\hat{S}_i} \times \boldsymbol{\hat{S}_j}) \tag{1.5}$$

where $\boldsymbol{D_{ij}}$ is the DMI vector accounting for the exchange interaction between the neighbor magnetic moments. Due to symmetry considerations of the structure the $D$ vector has a direction perpendicular to the plane hosting the two spins and the third high spin orbit element. Differently from the traditional exchange interactions, this term favors a rotation of neighbor spins. For this reason DMI interaction is also called anti-symmetric exchange. [5]

The DMI interactions can be found in bulk materials lacking inversion symmetry [6, 7] where the presence of a high spin orbit material determine this kind of interaction between neighbor magnetic moments. DMI can be found also in thin films composed by a high spin orbit metal and a ferromagnetic material [8, 9, 10].



(a) (b)

Figure 1.2: Dzyaloshinskii–Moriya interaction in bulk materials (a) and at the interface between a ferromagnet and an high spin orbit metal (b)

These two classes of material will be particularly important for the stabilization of skyrmion magnetic textures.

### 1.1.4 Demagnetizing energy

Demagnetizing energy originates from the direction of magnetic moments inside the sample. The demagnetizing energy is one of the non-local contributions to total magnetic energy. Knowing the direction of a magnetic moment and the direction of the demagnetizing field the energy density of the demagnetizing field can be computed as:

$$E_{demag} = -\frac{1}{2}\boldsymbol{M} \cdot \boldsymbol{B}_{demag} \tag{1.6}$$

where $boldsymbol B_{demag}$ is the demagnetizing field generated by the different magnetic moments of the material.

### 1.1.5 Anisotropy Energy

Anisotropy energy is a kind of energy that can determine together with demagnetizing field an anisotropic behavior between neighbor magnetic moments. In particular this kind of energy sets in different materials favorable directions for magnetization. Its origin can be found in the symmetry of atomic spins in the hosting lattice. The anisotropy arising from this interaction is called Magnetocrystalline anisotropy. The main interaction giving rise to such phenomena is the Spin Orbit Coupling (SOC) [4]. Depending on the symmetry of the lattice different directions can be favored. From an isotropic representation of energy like the one produced by the exchange interaction alone represented by a sphere, figure 1.3a, a possible anisotropic energy function is represented by the shape in figure 1.3b. The points of interest in such energy landscape are the one in which the directional derivative with respect to magnetization of the anisotropy energy density function is zeroed:

$$\frac{\partial f_{\mathbf{AN}}(\vec{m})}{\partial \vec{m}} = 0 \tag{1.7}$$

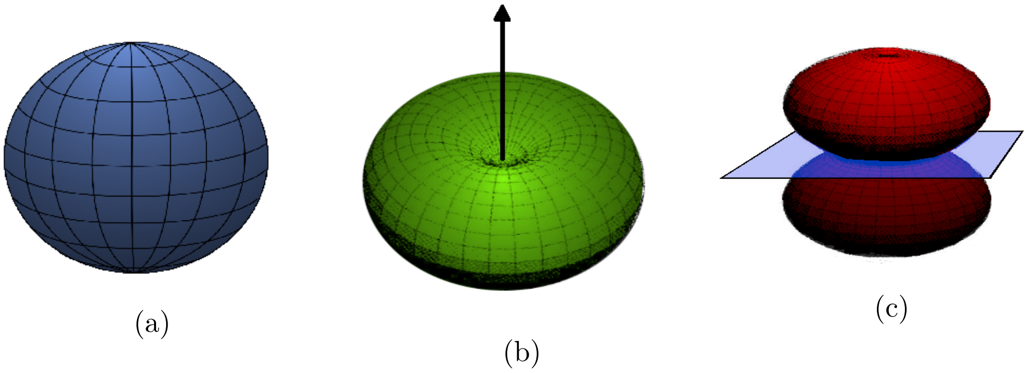
where $\vec{m}$ is the magnetization vector and $f_{\mathbf{AN}}(\vec{m})$ represents the anisotropy free energy function.

The local minima of the anisotropy density function are solutions of the above equation they identifies preferred directions in magnetization and are usually called *easy-axes*. The other solutions of the equation are saddle-points and local maxima, these points are usually referred as *medium-hard axis* and *hard axis* respectively.

The most simple representation of anisotropy energy is the one in which a single space direction dominates. In this kind of anisotropy the energy is invariant with respect to rotation around this direction and the energy is a function of the angle between the magnetization vector and this direction. This kind of anisotropy is called *Uniaxial Anisotropy*. A general expansion for the a typical energy density function describing the uniaxial anisotropy case is the following:

$$f_{\mathbf{AN}}(\vec{m}) = K_0 + K_1 sin^2\theta + K_2 sin^4\theta + K_3 sin^6\theta + ... \tag{1.8}$$

in which $\theta$ is the angle between the dominating anisotropy space direction and the magnetization vector $\vec{m}$, and $K_n$ are the anisotropy constant of order $n$. Taking into account only the first order in equation 1.8 two kind of anisotropy can be easily identified: *easy-axis anisotropy* and *easy-plane anisotropy*. In the first, $K_1$ is positive and the resulting energy density function is the one showed in figure 1.3b. As it is possible to see, the anisotropy axis in this case has become the preferred direction or *easy-axis*. In fact the energy density function has only two energy minima corresponding to $\theta = 0$ and $\theta = \pi$. In the second, $K_1$ is negative and the minimum of the $f_{\mathbf{AN}}$ function is obtained for $\theta = \frac{\pi}{2}$. This correspond to the magnetization lying on the plane orthogonal to the anisotropy axis. For this reason


(a)

(b)

(c)

Figure 1.3: Energy distribution as function of the magnetization direction a) Exchange interaction normalized energy distribution b) Uniaxial anisotropy energy distribution with positive first order anisotropy constant resulting in an easy axis anisotropy c) Uniaxial anisotropy energy distribution with negative first order anisotropy constant resulting in an easy-plane anisotropy

there is not anymore a preferred axis for magnetization, but a continuous degeneracy of equivalent orientations lying on a plane. The resulting energy landscape is represented in figure 1.3c. In addition to this form of anisotropy, non-isotropic behavior can be caused also by demagnetizing fields originated by grain or sample non-symmetrical shape and mechanical stress causing a forced movement of ions inside the material.[4] The first phenomena is referred as shape anisotropy [11] , due to the dependency from the shape of the magnetic sample, the second is called magnetoelastic anisotropy. [12]

### Interface magnetic anisotropy

Considering all the anisotropic contributions it is possible to express the magnetic anisotropy density with an effective anisotropy term $K_{eff}$ in the following form:

$$E = (K_{eff}) sin^2 \theta \tag{1.9}$$

The $K_{eff}$ term comprehend both the contributions from the anisotropic effect generated by the demagnetizing fields and from the magneto crystalline anisotropy.[13, 14]

In case of thin films with a thickness of few nanometers or lower, an additional term should be added to the effective anisotropy effect, the interface magnetic anisotropy contribution. In thin films with very low volume the interface presence promotes an additional term in the anisotropy energy.

Considering a ferromagnetic material sandwiched between two other non-magnetic layers the effective anisotropy can be expressed as:

$$K_{eff} = K_v + \frac{2 * K_s}{t} \tag{1.10}$$

where $K_s$ is the surface magnetic anisotropy term, $K_v$ is the term that includes the magneto crystalline anisotropy and the anisotropy generated by demagnetizing fields and mechanical stress and $t$ is the thickness of the ferromagnet. In equation 1.10, the ferromagnet is considered sandwiched between two equivalent materials. In case of different materials the term $K_s$ should be substituted by the two interface anisotropy constants specific for the two interfaces. It is important to notice that $K_s$, usually in the order of $1 \times 10^{-3}\,\mathrm{J\,m^{-2}}$, becomes comparable with the anisotropy term in the order of $1 \times 10^{5}\,\mathrm{J\,m^{-2}}$ in materials like Co or CoFeB, only for thicknesses in the order of nanometers. [13]

The critical thickness $t_c$ can now be defined as the negative ratio between the surface anisotropy and the other contribution.

$$t_c = -\frac{2 * K_s}{K_v} \tag{1.11}$$

At a thickness equal to $t_c$, the $K_{eff}$ term is 0 indicating a transition from a regime in which the $K_v$ term was dominating in the $K_{eff}$ calculation to another in which $K_{eff}$ is strongly affected by the surface induced anisotropy.

This concept is of great importance in the materials explored in this thesis that show, thanks to the surface contribution, a magnetization perpendicular to the magnet plane. In particular, skyrmions require the presence of a perpendicular magnetic anisotropy (PMA) of the material to be stabilized. Also the perpendicular nanomagnetic logic (PNML), discussed later in the thesis is based on single easy-axis anisotropy materials with strong PMA.

Finally the origin of the $K_s$ at the interfaces between a ferromagnet and an oxide can be found in the complex hybridization process between electronic orbitals at the interface between the two materials. It was also demonstrated both theoretically [15] and experimentally [16, 17] that applying an electrical field at the interface of between these materials can influence the strength of the magnetic anisotropy. This effect can be associated with the influence of the electrical field applied on the electron accumulation at the interface and the ion-migration from the oxide. This effect is called Voltage Controlled Magnetic Anistropy (VCMA) and it is of great interest for applications as will be discussed later in the text. [16]

### 1.1.6 Zeeman Energy

Zeeman energy is the energy linked to external magnetic fields. When an external magnetic field, $H_{ext}$ is applied to a magnetic media the related energy density can be computed as follows:

$$E_{Zeeman} = -\mu_0 M_{sat} \boldsymbol{m} \cdot \boldsymbol{H_{ext}} \tag{1.12}$$

### 1.1.7 Thermal Energy

The thermal energy is a term considered in the computation of the free energy of a magnetic system linked with temperature effect on the single magnetic moment. A possible formulation for such energetic component was given by Brown in [18]. The thermal fluctuation were derived by Brown for the case of a single domain magnetic particle fluctuating for the temperature effect. The thermal field can be modeled in the micromagnetic framework as a fluctuating field in the following form [19]

$$\boldsymbol{B_{therm}} = \boldsymbol{\eta} \sqrt{\frac{2\mu_0 \alpha k_B T}{\mu_0 M_{sat} \gamma_{LL} \Delta V \Delta t}} \tag{1.13}$$

where $\alpha$ is the damping constant, $T$ is the temperature, $k_B$ is the Boltzmann constant, $M_{sat}$ is the saturation magnetization, $\gamma_{LL}$ is the gyromagnetic ratio and $\eta$ is a random vector direction with normal distribution that varies in time. $\Delta V$

is the dimension of the unit volume used for the discretization and $\Delta t$ is the time step used to evaluate its variation.

This theory requires a more complex formulation of the LLG. The expressed temperature is often not referred to a real temperature but more often to an effective temperature especially when temperature is far from 0K. [20]

## 1.2 Skyrmions

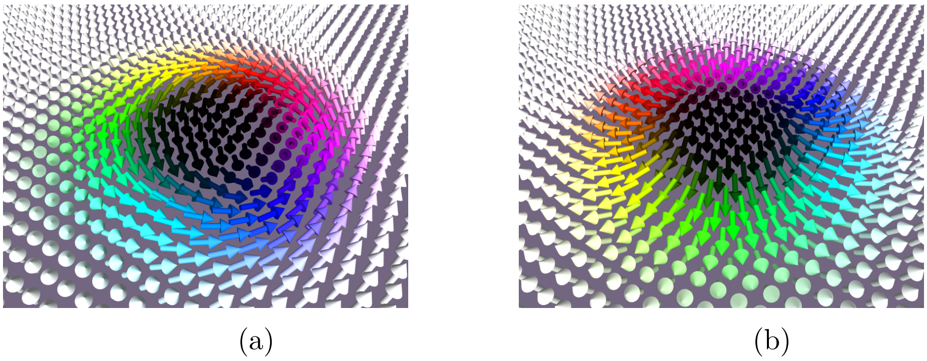
Skyrmion theory finds its origin in nuclear field theory. In 1962 the physicist Tony Skyrme proposed skyrmions as topologically stable field configuration of non-linear sigma models to describe interacting pions [21]. This concept was used then in other branches of physics to describe other phenomena as elementary particles, liquid crystals, quantum Hall magnets and Bose-Einstein condensates.[22] In magnetism, the concept was used to indicate a magnetic configuration that in the micromagnetic continuum approximation is represented by a topological soliton with well-defined characteristics. Magnetic skyrmions are defined as chiral spin textures with whirling configuration. These configuration predicted theoretically by Bogadnov et al. [22] were then experimentally demonstrated first in many types of B20-type bulk helimagnets [23] and later in thin films with heavy metals interfaces [10, 24, 25]. Particularly appealing for applications in solid-state circuits was the demonstration of skyrmions stabilization at room temperature and low or zero bias field.[**boulle2016room**]

### 1.2.1 Magnetic skyrmions types

The main interaction that allows to stabilize skyrmions in ferromagnetic systems is the DMI, discussed in section 1.1.3. This interaction favors in fact the spin canting allowing the existence of helical and skyrmion states. [26] Depending on the different kind of DMI, different types of skyrmions are stabilized in magnetic materials. In presence of bulk DMI, interaction present in B20 ferromagnets, the skyrmion stabilized is called Bloch skyrmion. This kind of skyrmion is shown in figure 1.4a. In case the DMI interaction is developed at the interface between the ferromagnet and another material, the stabilized skyrmion is called Néel Skyrmion. This case is common in thin films where the ferromagnetic material is in contact with an high spin orbit material.

Both the skyrmions are characterized by a core polarization opposite to the one of the bulk. The magnetization goes from the core to the outer polarization by means of a domain-wall, in which the magnetization rotates gradually from one polarization to the other.

Bloch skyrmions are characterized by a domain wall, enclosing the structure,



(a)

(b)

Figure 1.4: Skyrmion textures. a) Bloch type skyrmion; b) Néel type skyrmion

with circular chirality. In this case the magnetization, from the core, rotates clockwise or counter clockwise in the plane tangent to the skyrmion profile. This transition produce a skyrmion with spiral-like shape. Néel skyrmions differ from the previous in the transition zone. The transition from the inner polarity to the external is a rotation in the plane perpendicular to the skyrmion profile. The rotation can happen also in this case in both clockwise and counterclockwise direction.

### 1.2.2 Topological charge

In most definitions, magnetic skymion are characterized by a set of numbers that allows a compact description of this magnetic texture. These three numbers are the topological charge, the helicity number, and the vorticity number. The most important number in the definition of a skyrmion is the topological charge. This number is defined as follows:

$$Q_s = \frac{1}{4\pi}\int_S \vec{m}\cdot(\frac{\partial\vec{m}}{\partial x}\times\frac{\partial\vec{m}}{\partial y})dxdy \tag{1.14}$$

where $Q_s$ is the topological number, $\vec{m}$ is the normalized magnetization vector and $S$ is the ferromagnet surface. For skyrmion textures this number, referred also as Pontryagin number, is an integer number. Geometrically, the absolute value of the topological number equals to the number of times the magnetic texture wraps the unit sphere. The skyrmion can be mapped to a sphere considering one spin direction at one pole and the boundary spins at the other and mapping all the other spin around the sphere.

### 1.2.3 Vorticity number

Taking advantage of the symmetry of skyrmions, it is possible to express the magnetic texture of skyrmions in this form [27]:

$$m(r) = (cos\Phi(\phi)sin\Theta(r), sin\Phi(\phi)sin\Theta(r), cos\Theta(r)) \tag{1.15}$$

where polar coordinates $\mathbf{r} = (rcos\phi, rsin\phi)$ were used. In particular the functions $\Phi$ and $\Theta$ represent respectively the polar and the azimuthal angle of the magnetic moment.

Now, the 1.14 can be rewritten in the following terms:

$$Q_s = \frac{1}{4\pi}\int_0^\infty \int_0^{2\pi} \frac{d\Theta(r)}{dr}\frac{d\Phi(\phi)}{d\phi} sin\Theta(r) dr d\phi = \frac{1}{4\pi}[\Theta(r)]_{r=0}^{r=\infty}[\Phi(\phi)]_{phi=0}^{phi=2\pi} \tag{1.16}$$

The term $\frac{[\Phi(\phi)]_{phi=0}^{phi=2\pi}}{2\pi}$ is defined as the vorticity number. More in general the vorticity can be defined as the number of revolutions a vector does with reference to a defined direction following a path. In case of skyrmions, the number expressed by the vorticity refers to the rotation of the projection on the xy plane of the magnetic moments constituting the skyrmion texture considering the z axis as the direction normal to the film and moving along the texture profile in a closed path.

In figure 1.5, it is possible to see different vorticity calculations for different magnetic textures.

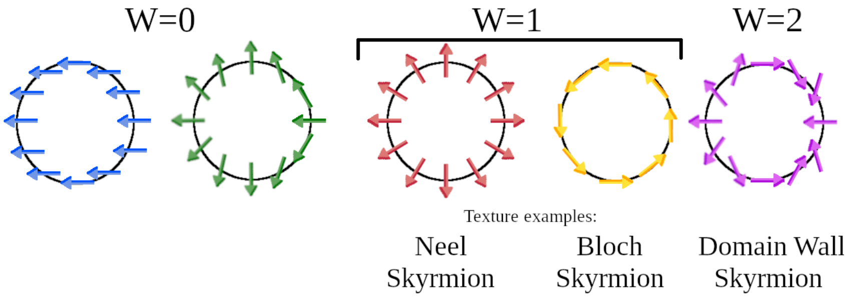


Figure 1.5: Vorticity computation for different magnetic configurations.

Finally computing the vorticity number, $Q_v$, it's possible to compute the skyrmion number:

$$Q_s = \frac{Q_v}{2} * [\Phi(\phi)]_{phi=0}^{phi=2\pi} \tag{1.17}$$

### 1.2.4 Helicity number and polarity

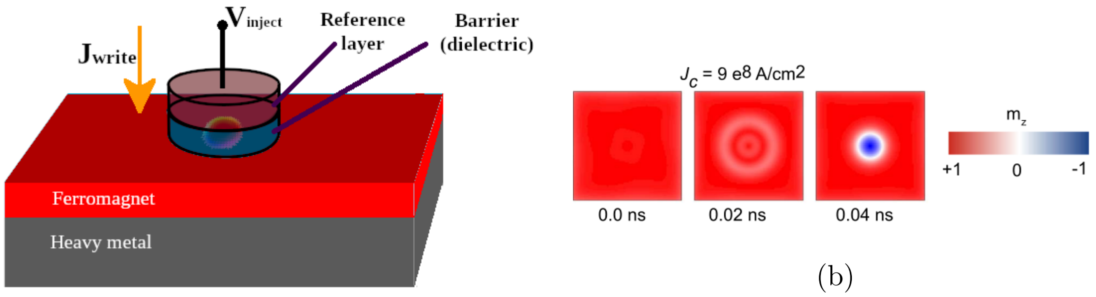
The helicity number, $Q_h$, is the third parameter used to characterize the skyrmion. The helicity number is defined as the phase in the equation of the azimuthal angle of magnetization [27]:

$$\Theta(r) = p\phi + \gamma \tag{1.18}$$

where $\gamma$ is the helicity number, $p$ is the core polarization of the skyrmion and $\phi$ is the azimuthal angle. This parameter is determined from the kind of DMI interaction that stabilizes the skyrmion in the magnetic system. In case of bulk DMI, that stabilize a Bloch skyrmion, the helicity number will be $\frac{\pi}{2}$ or $\frac{3\pi}{2}$. In case of interfacial DMI, the the helicity number will take a value equal to 0 or $\pi$. Finally the polarization $m$ is determined by the direction of the magnetization in the core of the skyrmion.

### 1.2.5 Nucleation

Nucleation of skyrmions is a crucial mechanism for applications. An important characteristic of a skyrmion device is the ability to nucleate a skyrmion depending on the input information. This characteristic is also important when considering the advantages of the realization of skyrmion memories as data storage devices. The principal mechanisms shown in literature are based on different phenomena temporal variation of local anisotropy [28], external magnetic fields [29], current injection and nucleation with magnetic tunnel junctions [30]. The nucleation by means of vertical spin polarized currents was one of the first proposed in literature [31]. The nucleation is based on the injection of a vertical polarized spin current. The polarization of the injected current is in the direction of the skyrmion core. In this process, the injection favors the nucleation of a new domain that, then, is stabilized in a skyrmion state. The nucleation of skyrmions with this process has been studied theoretically and through micromagnetic simulations. The results of simulations show how the amplitude of the pulse should be carefully calibrated to avoid the nucleation of generic domains or the missed creation of the skyrmion. A favoring condition can be also represented by the presence of an external field that lowers the nucleation time of the skyrmion. [31, 32]



(a)

(b)

Figure 1.6: Nucleation of skyrmion by means of vertical polarized current in a circular dot of CoFeB $\alpha = 0.015$, $D$ =3 mJ m$^{-2}$. Image reproduced from [31]

Another mechanism proposed for controlled nucleation is the use of injected currents in the ferromagnet plane. The current injection in the magnet plane can be used in two different ways to produce a skyrmion: to favor the topological transition of a domain or by nucleation at defects. The first technique proposed by Zhou et al. in [33] and by Jiang et al. in [34] takes advantage of the current to push a generic domain in a constriction and to generate from it a skyrmion texture. The mechanism recalls the one of generation of soap bubble or the generation of a droplet from a liquid. Finally a similar method was also employed for the realization of a nano-device by Finizio et al., where a vertical contact at first nucleates a domain from which a skyrmion is detached with the application of an in-plane current [30]. Another method that takes advantage of in plane current was proposed by Iwasaki et al. [35] and takes advantage of the spin rotation around an engineered notch to generate a bubble upon the application of a strong enough current. The mechanism has been studied in bulk materials where Bloch type skyrmions are stable configurations.

The usage of external magnetic field is also possible for the nucleation of skyrmions with magnetic field gradients as demonstrated by Casiraghi et al. [36], in which upon the application of a localized magnetic field by means of local magnetic field by means of a MFM tip skyrmions were nucleated. An external magnetic field was also demonstrated to be capable of nucleating a skymion in presence of an engineered defect. In [29] Fallon et al. demonstrated how an engineered defect, created by means of focused ion beam irradiation, can produce locally a skyrmion in the position of the defect upon the variation of the applied magnetic field.

Finally a method that was demonstrated recently is the nucleation obtained with local anisotropy control by means of VCMA. In [28] the authors reported the creation of skyrmion textures upon the application of a voltage to an heterostructure with antiferromagnet/ferromagnet/oxide composition. The electrical control of anisotropy was demonstrated effective also for the manipulation [37] and movement of skyrmions in tracks [38].

### 1.2.6 Detection

The detection of skyrmions, as well as the nucleation is a fundamental mechanism in applications. The skyrmion devices need the capability of communicating to the other systems the information they encode.

In particular, the detection is possible by optical and electrical means. The optical methods are suitable for laboratory environments where the skyrmion phenomena are isolated and studied. Different imaging techniques are available. Among them one of the most used is the Magneto-Optical Kerr Effect (MOKE) imaging that is based on the optical Kerr effect. The magnetic polar Kerr effect links the change in polarization or intensity of incident light with the magnetization of the imaged material. In particular the polarization change of the light is widely used to image magnetic materials. The limitation of this technique is in the resolution that is limited by the diffraction limit of the visible light. For this reason, often skyrmion structures that can have dimensions in the order of hundreds or tenth of nanometers are often imaged also using X-ray magnetic circular dichroism (XMCD) technique. In particular the XMCD is the effect correspondent to the MOKE effect for X-rays [39], the technique measures the absorbtion of X-ray polarized beam and uses this information to build an image of the magnetization. The lower wavelength of X-rays with respect to visible light used in MOKE measurements, allows to image magnetic textures with higher resolution and has been used in different studies to image skyrmions in ferromagnetic materials. [40, 41, 42].

Despite these techniques give important information on skyrmions, they're only suitable for experiments. The electrical detection is indeed required for application. Regarding electrical detection of skyrmion three main techniques have been proposed: Hall effect [43, 44], resistance change in magnetic tunnel junction with change in the tunneling magneto resistance [45, 46] and non-collinear magneto resistance effect [47]. These techniques represent a great promise for the integration of circuits based on magnetic skyrmions. In particular the reading by means of magnetic tunnel junctions is promising for its compactness. At the same time all these techniques showed low signal to noise ratios that makes the electrical detection of skyrmion a challenging task with many open challenges.

### 1.2.7 Skyrmion dynamics

One of the most important characteristic of skyrmions is their response to current stimulus. In fact when subjected to current, the skyrmion, like other magnetic textures, moves showing a rich dynamic. During the movement, up to a current density threshold, the skyrmion drifts. This movement suggested skyrmions as possible alternatives for Domain wall (DW) racetrack memory proposed by Parkin in 2008 [48, 21]

#### Spin Transfer Torque

A phenomena that can promote the movement skyrmions in ferromagnetic material is the STT. This effect can be present in different scenarios. For example when a current passes through a ferromagnetic layer after crossing another one with magnetization non-collinear. In this case, the second ferromagnet absorbs part of the angular momentum that is carried by the electrons. Another case in which spin transfer torque is present is when a current crosses a non-uniform magnetic configuration like a domain wall. In both scenarios, the local angular momentum of the magnetic material is transferred to the local magnetic moments. Therefore, the local magnetization feels an equal opposite torque. This contribution has been inserted in LLG equation adding two terms to the torque: an adiabatic torque and a non-adiabatic one. The first torque refers to situations in which the transfer of momentum is complete and the spins follow adiabatically the local magnetization.

This torque can be expressed with the following equation [49]

$$\boldsymbol{\tau_{STT-ADIABATIC}} = \frac{\mu_0 \gamma \hbar P}{2\mu_0 e M_s} (\boldsymbol{j} \cdot \boldsymbol{\nabla}) \boldsymbol{m} \tag{1.19}$$

where $\gamma$ is the gyromagnetic ratio, P is a coefficient of polarization, $e$ is the electron charge, $M_s$ is the saturation magnetization, $j$ is the current density flowing in the ferromagnet, $m$ is the magnetization direction.

In systems with non-negligible spin-orbit coupling the picture is not complete with the term presented in the equation 1.19. For this reason a second term is taken into account and is present in systems in which the variation of magnetization is rapid and the spin of electrons crossing the magnetization cannot follows completely. This non-adiabatic contribution is expressed with the following expression:

$$\boldsymbol{\tau_{STT-NON-ADIABATIC}} = \frac{\mu_0 \gamma \hbar P}{2\mu_0 e M_s} \beta \boldsymbol{m} \times (\boldsymbol{j} \cdot \boldsymbol{\nabla}) \boldsymbol{m} \tag{1.20}$$

where $\beta$ is the non-adiabatic constant.

### Spin Hall Effect

The Spin Hall Effect is a phenomena involving generation of transversal pure spin current in response to charge current traversing a non-magnetic layer. In particular the non-magnetic layers generating the spin currents are characterized by high spin orbit coupling like heavy metals. In the spin Hall effect a spin accumulation with opposite spin polarity happens at opposite edges of the metallic layer. In a stack with a ferromagnet in contact with the non-magnetic metal, the spin accumulation produced by a current flowing in the non-magnetic layer expands in the ferromagnet. A pure spin polarized current flowing in the vertical direction is then produced. This current interacts with magnetization producing a torque on the local magnetization. The effect of spin Hall current can be included in LLG with a torque term in the following form:

$$\tau_{SHE} = -\frac{\gamma_0 \hbar j \theta_{sh}}{2\mu_0 e M_s d} \boldsymbol{m} \times (\boldsymbol{m} \times \boldsymbol{p}) \tag{1.21}$$

where $p$ is the polarization direction of the current, $\theta_{SH}$ is the spin-hall angle and $d$ is the thickness of the ferromagnetic layer. The polarization of the spin current can be computed as $(\hat{z} \times j_{HM})$ where $\hat{z}$ is the direction normal to the ferromagnet plane. The spin Hall angle indicates the ratio between the current flowing in the heavy metal and the spin current generated.

### Skyrmion motion under current stimulus

When subjected to current stimulus, skyrmions in ferromagnetic systems show both a longitudinal and a transverse velocity with respect to the current direction.

The presence of both the components makes the skyrmion deviate on one side of the sample while moving in the direction of the current. This effect is called Skyrmion Hall Effect (SkHE) and is responsible for the upper current threshold showed by skyrmions that moves in a nanowire. [24, 50]. To better understand the current movement in response to a stimulus a formalism that can be used is the Thiele formalism. This formalism was developed by Thiele in 1973 to describe the movement of hard magnetic bubbles and other domains in steady movement [51]. Considering a stable magnetization and a steady state movement the movement of the skyrmion can be expressed with the equation 1.22. The following equation refers to a skyrmion pushed by a current flowing in the underlayer of a thin film ferromagnet. The driving effect is therefore the Spin Hall Effect (SHE). The considered skyrmion texture is a Néel skyrmion [52]

$$\boldsymbol{G} \times (\boldsymbol{v_d}) + \alpha D\boldsymbol{v_d} + 4\pi B\boldsymbol{J_{HM}} + \boldsymbol{F} = 0 \tag{1.22}$$

where $v_d$ is the drift velocity of the skyrmion center, G is the gyromagnetic vector and finally $B$ is a factor used to convert convert the electrical current stimulus in the drift force for the skyrmion movement [52]. The term $F$ takes into account all the other forces that derive from other characteristics of the mean in which the skyrmion is moving as the confinement effect, the local variations of anisotropy, the repulsion from other skyrmions and others.

The gyromagnetic vector $G$ for a single skyrmion can be expressed as $G = (0{,}0{,}4\pi Q)$ where Q is the skyrmion number. This component in particular is the source skyrmion Hall effect. The factor D is the dissipative tensor. The dissipative factor $D_{ij}$ can be computed with the following [24, 52]

$$D_{ik} = \frac{1}{4\pi}\int_S (\frac{d\boldsymbol{m}}{dx_i} \cdot \frac{d\boldsymbol{m}}{dx_k}) dx_i dx_k \tag{1.23}$$

For magnetic skyrmion $D_{ik} \sim 1$ for $i = k$ and $D_{ik} = 0$ for $i \neq k$.

From the above equation, considering a current flowing in the x direction $\boldsymbol{J_{HM}} = (j{,}0{,}0)$ and the additional potential $F = 0$, the following velocities can be expressed explicitly:

$$\begin{cases} v_{d,x} = \frac{-j\alpha DB}{(\alpha D)^2+Q^2} \\ \\ v_{d,y} = \frac{jQB}{(\alpha D)^2+Q^2} \end{cases} \tag{1.24}$$

From the above, it is possible to see that as long as a current is present both the components are present and that the ratio $\frac{v_{d,x}}{v_{d,y}} = \frac{-Q}{\alpha D}$. Being $D =\ 1$ and $\alpha < 0.1$ for a common ferromagnetic material, it is possible to see that $v_{d,y} >> v_{d,x}$. In common ferromagnetic material, the transverse velocity is higher in magnitude with respect to the drift velocity in the current direction. [24]

Finally, it is worth noticing that with different core directions, equivalent to skyrmion numbers $Q$ with opposite signs, the transverse velocity $v_{d,y}$ changes sign, while the drift velocity in the current direction remains in the same direction.

# Chapter 2

# Design flow for complex circuits based on skyrmions

## 2.1 Motivation

The first goal of this thesis is to bridge the gap between the device level and complex systems simulations based on skyrmion technology. The first has been used extensively in literature and it's based on a well established approach, that is the micromagnetic approach [19, 53, 54], section 1.1. This simulation level is usually solved using a finite difference or finite element approach. Many software have been proposed for the numerical solution of this problem but among them the most used are OOMMF [54], the de facto standard for micromagnetic studies and Mumax3, a GPU-accelerated micromagnetic finite-difference software [19]. Both OOMMF and Mumax3 solve the micromagnetic problem using a finite-difference approach. This approach requires a meshing step of the device under study, respecting the characteristic exchange length of the material defined as $l_{ex} = \sqrt[2]{\frac{A}{\frac{\mu_0 M_s^2}{2}}}$, e.g.5.6 nm for CoFeB thin films [52]. At this point, the simulator computes the torque for every element of the mesh using an adequate form of the LLG equation and advance of a single time step. This process is then repeated until the running conditions, the total simulation time or other physical conditions, are met and then the simulation is stopped. The great number of independent computations required, made the employment of GPU devices particularly effective in the solution of micromagnetic problems [53]. This approach allows to describe with sufficient precision a great number of magnetic phenomena and it's widely used to explore both the basic physics and applications in devices. Part of this thesis, as well, is based on this approach in order to verify at device level the correctness of the proposed solutions. Despite the great precision and efficiency given by this approach, this method is difficult to apply to complex systems. These systems, based on hundreds of gates, exceed by far the acceptable dimensions for micromagnetic simulators. Even restricting the study to basic blocks, the computational effort remains considerable and consequently the time required for simulations. An example of the time required for simulations is shown in table 2.1. Structures with increasing complexity were simulated with mumax3 measuring the run time of the simulation to perform a 5 ns simulation. The simulation were performed on a NVIDIA Tesla P100. The mesh size is set to $1\,\mathrm{nm}^2 \times 1\,\mathrm{nm}^2 \times 0.5\,\mathrm{nm}^2$. The simulations were run with adaptive time step with minimum value $1 \times 10^{-14}$ ns and maximum value $1 \times 10^{-12}$ ns with solver rk45, employing the Dormand-Price method for the solution of the LLG equation. [19]

Table 2.1: Computation time required by mumax3 on GPU for 5 ns of simulation

| **Structure** | **Mesh Size** | **Simulation Run Time** [s] |
|---|---|---|
| Simple Track | $512 \times 30 \times 2$ | 7 min |
| Curbed Nanotrack | $512 \times 50 \times 3$ | 16 min |
| AND/OR gate - Skyrmion Logic [55] | 150x200x3 | 15 min |
| XOR gate - Skyrmion Logic | 512x256x3 | 48 min |

When dealing with devices designed and verified with micromagnetic simulations, a possible design methodology of logic systems is the one showed in figure 2.1.

The presented design flow represents a possible approach to design a logic system with part of the circuit realized in skyrmion technology and part in CMOS. In logic circuits based on emerging technologies usually a part of the logic is realized in transistor technology to complement the functions that cannot be easily realized with that technology or for compatibility with the electrical environment . In the architectures that will be analyzed later in the thesis the control is usually realized in CMOS technology for the superior speed and communication capabilities of this technology with respect to skyrmions. In the presented design flow the first step is the identification of the basic operations and of the basic boolean logic structure of the circuit, used to perform the algorithm. With these information, the design process proceeds to the technology dependent phase. The design procedure focuses on one side on the design of the datapath for data elaboration and on the other on the design of the control unit required for the correct operation of the datapath. The control unit, usually designed in CMOS technology, can now follow a well defined path, starting with a VHDL or Verilog description of the circuit that will be used for its simulation and later for the synthesis. The toolchain used for this design is standard in industry. On the contrary, the datapath needs the definition

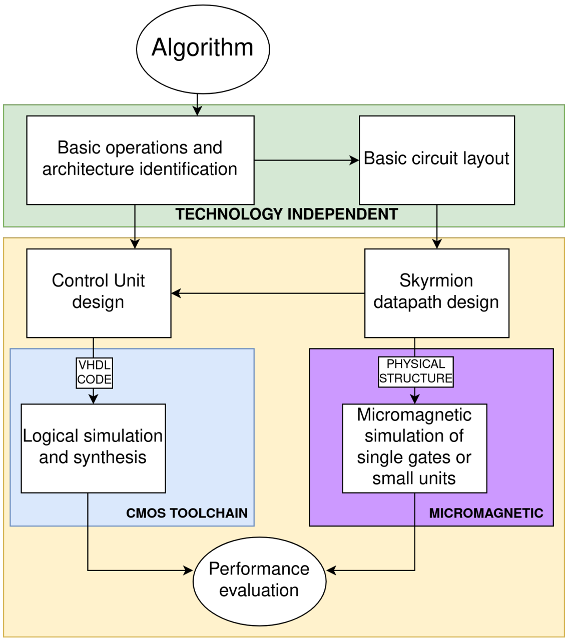


Figure 2.1: Design flow example using exclusively micromagnetic simulation for the datapath design

of the physical structure of the required gates and interconnections required for the correct circuit functionality. The physical circuit needs now to be simulated. If the circuit is extremely regular the simulation of basic gates is enough to represent the complete circuit functionality. In case of circuits composed by many different units and with a complex interconnection schemes, many simulations of the different parts of the circuit are needed. In addition, for every block, all the admitted input combinations should be tested. As previously shown in 2.1, a basic block as a XOR gate can require up to 48 minutes per input combination for a 5 ns simulation. For skyrmions logic, in addition the gates can have additional electrical inputs used to control the skyrmion flow in the gate, that increase the number of conditions to test in the circuit. After the basic blocks have been simulated, the performance can be extracted from the simulations and joined with the results from the logical synthesis to derive the overall performance of the design. In this situation the design flow presented above misses two important steps that can be critical in the design of complex circuits: only small portion of the skyrmion datapath can be executed at once and the control cannot be verified directly on the complete circuit. The former limits the verification possibilities of circuits in an acceptable time, a design error in this framework could lead to a non-negligible additional time in the design flow. The latter is fundamental to understand if the control can correctly guide the datapath with the correct timing. Finally, in this scenario even the verification of the interactions between gates is left to a manual verification process that can be difficult in case of complex circuits with long interconnections and lead to errors requiring expensive redesign and verification passages. In this chapter, an alternative to the previous design flow focused on skyrmion technology is proposed. In the proposed flow, the simulation of heterogeneous designs is executed in a single environment at the price of a reduction in the accuracy of the description of the phenomena involved but with a considerable speedup of the simulation time, an easier design process and the possibility to directly verify the complete circuit operation in a single environment.

## 2.2 Skyrmion devices

Before going into details on the design flow, a presentation of the state of the art regarding skyrmion gates for boolean logic is done. The solutions presented in this section are able to elaborate a boolean information stored in form of skyrmion encoded in presence of the magnetic texture and produce at the output an information in form of skyrmions. In many of the presented proposals the skyrmion value is read and the corresponding electrical signal is then used for further elaboration.

### 2.2.1 Skyrmion Domain-Wall logic

The first family of gates analyzed, was proposed by Zhang et al. in [56]. The logic proposed takes advantage of the possibility of converting a skyrmion into a domain wall and viceversa as already studied by Zhou et al. in [33]. The conversion from skyrmion to domain wall is based on the fact that an isolated domain colliding with an edge can break the continuity of the domain wall that encloses it. If the domain collides with the boundaries in a constriction small enough to host a domain, the conversion is possible as shown in figure 2.2. The opposite conversion is more delicate but it's based on the principle that in case a skyrmion is a stable texture in a material and the line width is enough to elongate the initial domain without collisions, a metastable domain can reduce to a stable skyrmion. This process as explained in [33] involves a a careful control of the geometry and of the current stimulus. Given the right material and geometry conditions, a wrong pulse can lead to the generation of a meron or another non-stable domain configuration and lead to the annihilation of a skyrmion. In case the stimulus is correct as shown in figure 2.2 the conversion is successful. Zhou et al. demonstrated this principle using STT for domain movement but the same, with some adjustments, applies with a stimulus generated with SHE as will be shown in section 5. Moreover some experiments from Jiang et al. [34] demonstrated experimentally a skyrmion generation phenomena starting from domains traveling in a constriction.

The gate family proposed by Zhang works taking advantage of this conversion. The skyrmion is translated into a DW pair and then pushed into a patterned

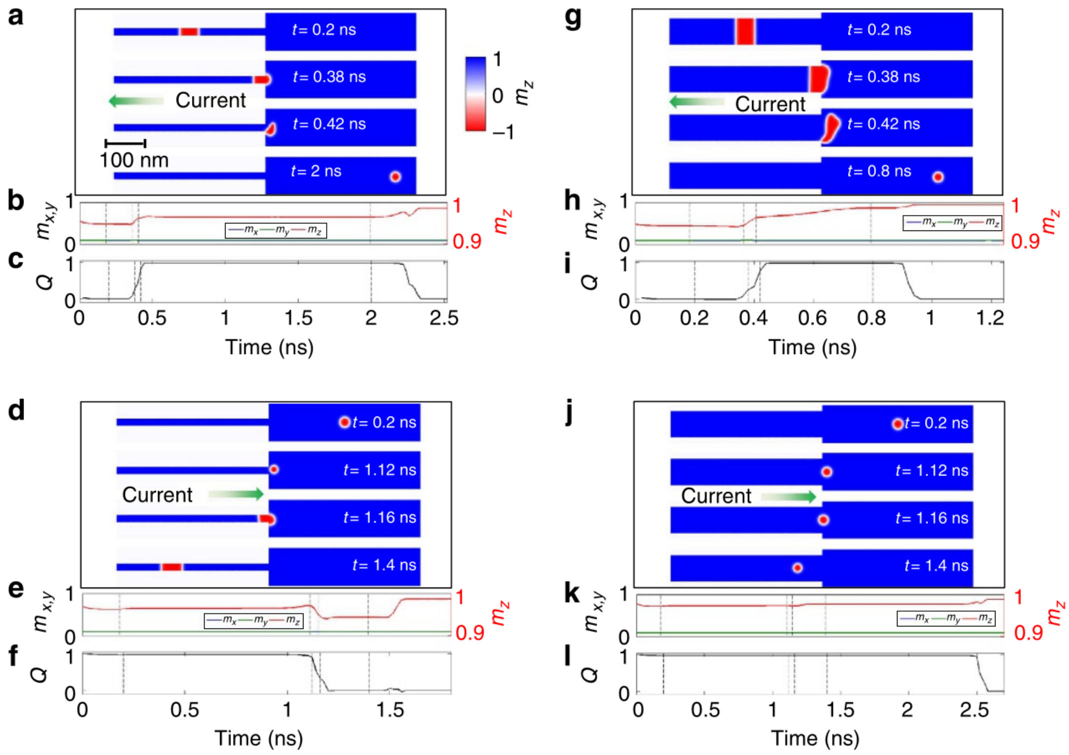
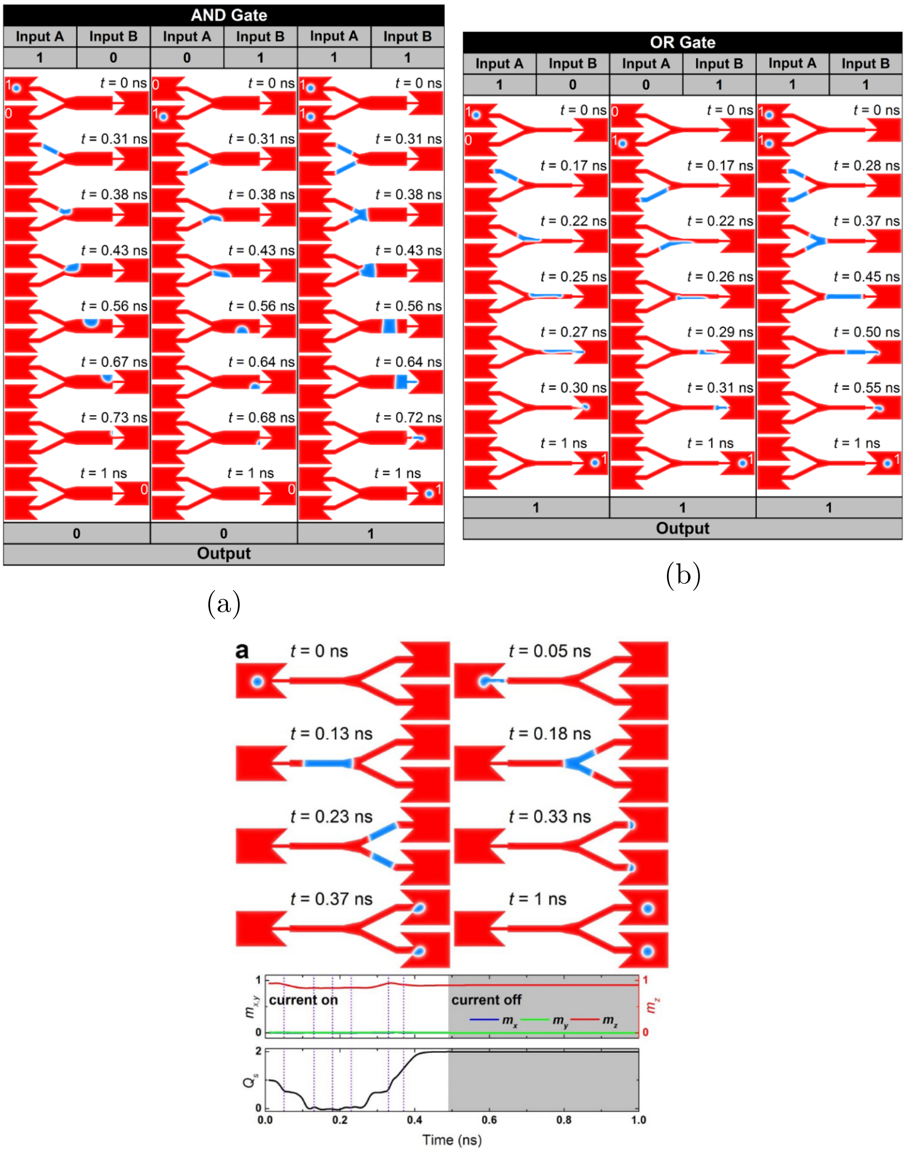


Figure 2.2: Skyrmion domain wall conversion. The figures have been extracted from [33] shows the conversion from domain to skyrmion (a-c,g-i) and the opposite process (d-f,j-l). The effect is guided by the STT effect.

structure that realize the logic function and finally is converted back in skyrmion for elaboration. This family of devices includes a duplication gate, an AND gate and an OR gate as shown in figure 2.3.

In the duplication gate as shown in figure 2.3c, after the first skyrmion-domain wall pair conversion, the newly generated domain is pushed by means of an electrical current into a fork shaped structure where it is split in two domains. They will be then converted back into skyrmions. The stability of the domain moving in the constriction makes this process particularly interesting for applications where duplication of information is required. The proposed OR gate uses the same structure of the duplication gate in the opposite direction, figure 2.3b. The two input skyrmions converted in two domains are pushed in the output constriction where in the limited space the two skyrmions are joined. The small width of the output constriction allows the survival of the domain even if only a skyrmion was present at the input. This function is logically equivalent to an OR operation. Finally, the AND gate presents the same structure as the previous gate, with the only difference of the central constriction that in this case is wider, figure 2.3a. The working principle can be derived from the one of the OR gate: the wider output constriction



(a)

(b)

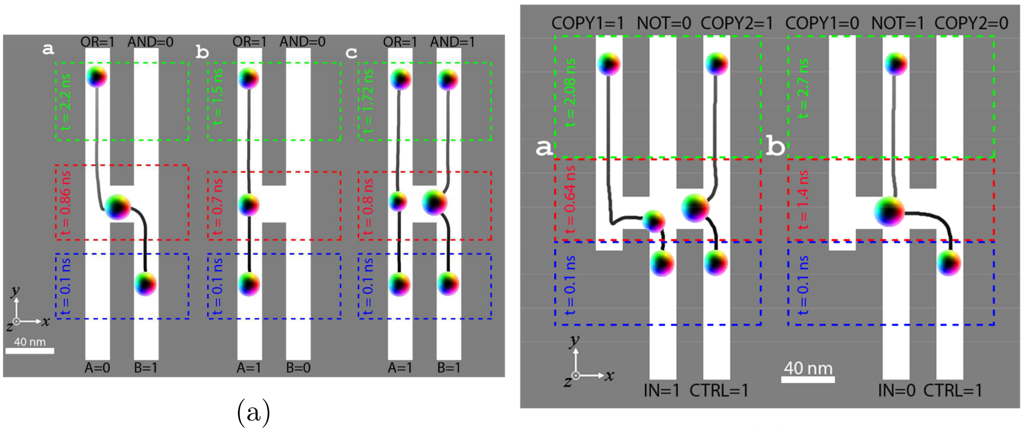
(c)

Figure 2.3: Skyrmion domain wall logic. a) AND gate b) OR gate c) Duplication mechanism based on domain wall. Images extracted from [56]

allows the formation of a stable domain at the output only in case two input domains reach the output line at the same time. The device family proposed by Zhang et al. is not a complete logic family and it is not based exclusively on skyrmions: the elaboration happens between plain domains. Nevertheless, the proposal gives an important contribution: the possibility to convert skyrmion in other magnetic textures opens interesting scenarios where different class of gates can be used on the same material stack, moving the information from one state to another. This concept will be useful in chapter 8 in defining interfaces between different magnetic technologies.

### 2.2.2 Skyrmion gates based on skyrmion-skyrmion and skyrmion-edge interaction

Another interesting proposal of logic gates was done by Chauwin et al. in [55]. The gates proposed takes advantage of the edge and the mutual repulsion of skyrmions to execute logic functions. An important feature of this gates is that no conversion is needed for information elaboration and that cascading of logic gates is trivial with respect to other solutions. The boolean logic gates proposed by Chauwin et al. are shown in figure. 2.4



(a)

(b)

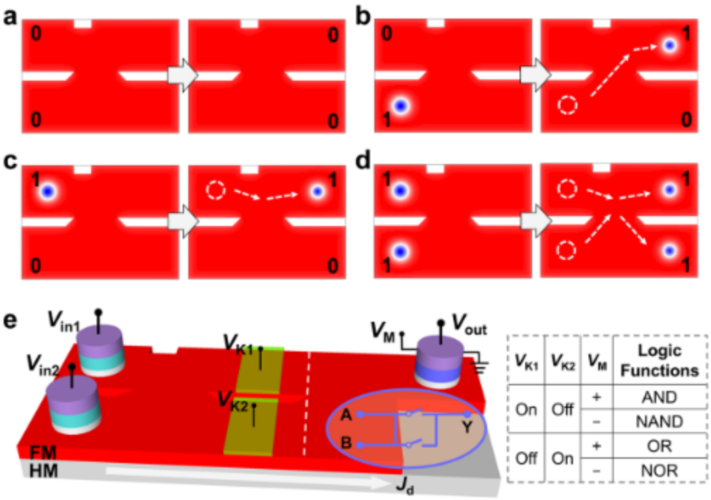
Figure 2.4: Skyrmion logic system based on skyrmion-edge and skyrmion-skyrmion repulsion effects. Images extracted from [55]

The first gate, shown in figure 2.4a, executes at the same time the AND and the OR operation on the input data. The information as the previous gate encodes the two logic states in the presence or absence of the skyrmion at the input position. The operation between the two inputs is executed when the skyrmions reach the central section of the logic gate. The skyrmion in the right track due to the skyrmion hall effect tends to deviate on the left. In case the second skyrmion is present, a logic 1, the first skyrmion is repelled and continues its motion in the right track. In case the skyrmion on the left is missing the first skyrmion changes track, ending in the left output. The result produced at the output of the right track is equivalent to the result of the logic AND between the two inputs. The result produced on the left is equivalent to the logic OR between the two inputs. A skyrmion will be always present at the left output if at least one of the two inputs is one. The second boolean gate proposed is shown in figure 2.4b. This gate realize the NOT function. The gate, to generate its functionality uses an additional skyrmion at the input. The input of the gate is on the central track. The function, as in the previous gate, is executed in the central section where the auxiliary skyrmion moves on the left in the direction of the central track. If the input skyrmion is present, the auxiliary skyrmion is repelled and continues its path in the rightmost track, otherwise, the auxiliary skyrmions continues its motion in the central track. The input skyrmion ends always in the leftmost track. This interaction between the input and the auxiliary skyrmion produces at the central track the equivalent of a NOT operation. In addition a copy operation is realized on the other two lines. The information at the input of the gate needs to be synchronized to correctly execute the logic functions. This synchronization can be realized using notches in order to stop the motion of skyrmions traveling in the tracks [55, 21] or VCMA gates. The two mechanism will be described more in depth later in section 2.2.4

Another proposal for gates based on edge and mutual skyrmion repulsion has been also proposed from Luo et al. in [57] and is shown in figure 2.5. The basic principle of skyrmion interaction is similar to the one proposed in the AND/OR gate shown above, where the inputs, encoded in skyrmions, interact via the mutual repulsion at the central section and modify their trajectory accordingly. The main difference here is that the gate is thought as a standalone unit where the input is nucleated at the input write head of the gate and read out at the end. In particular, also the read head in the gate plays a role with respect to the logic function realized by the gate, underlying the necessity of reading out the result to complete the elaboration of information. An important feature showed in some of these gates is the use of electrical inputs to modify the movement of skyrmions crossing the structure. In particular the reconfigurable gate proposed in this article implements different logic functions depending on the local modification of anisotropy obtained by means of two VCMA gates positioned at the center of the device structure and the control of the output read head.

### 2.2.3 Skylogic

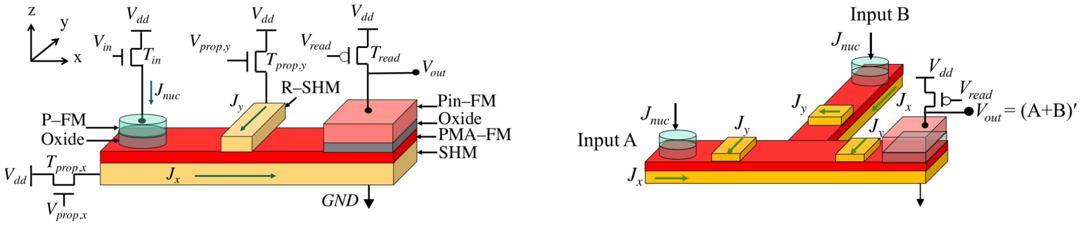
Another proposal for logic was done by Mankalale et al. in [58]. The gates are based on a track with an additional contact perpendicular to the track to counteract the deviation of the skyrmion toward one of the borders. The function realized from the logic is based on the way in which the read head placed at the end of the track reads the skyrmion. The NOT gate is realized by a track with a write head on one side and a read head on the other, figure 2.6a. The input, encoded in a skyrmion, travels from one side to the other of the gate pushed by a charge current flowing in the heavy metal layer underneath. When the skyrmion reaches the output read head the inverse orientation of the reference layer with respect to the bulk magnetization produce an high voltage in case a skyrmion is absent



| $V_{K1}$ | $V_{K2}$ | $V_M$ | Logic Functions |
|---|---|---|---|
| On | Off | + | AND |
| | | - | NAND |
| Off | On | + | OR |
| | | - | NOR |

Figure 2.5: Reconfigurable skyrmion logic gate. The control for reconfigurability is realized by the two VCMA gates guided by $V_{K1}$ and $V_{K2}$ and by the $V_M$ value controlling the output read head. Image extracted from [57].

and a low voltage in case a skyrmion is present. This voltage can now be used to drive other gates connected to the output. The gate, differently from the previous proposals, needs the read operation to realize its logic function. Without this last step the function is not realized. The NOR function is realized with a junction of two lines. Like the previous structure, both the lines are equipped with a write head that will produce a skyrmion, according to the input of the gate. The two inputs are then pushed toward the output read-head placed on the other side of the gate. The read head as implemented in the gate before produce a low-voltage, a logic 0 if a skyrmion reach the read-head or a logic 1, if no skyrmion is present under the read head. The two gates are shown in figure 2.6.



(a)

(b)

Figure 2.6: Skylogic gates: a) NOT logic gate b) NOR logic gate. The $V_{in}$ voltage signals produce produce the input for the gate. The $V_{propx,y}$ signals control the movement of the skyrmion. At the output the $V_{out}$ signal is the output of the gate. Image extracted from [58]

### 2.2.4 Skyrmion gates synchronization

The skyrmion gates presented in section 2.2 and 2.2.1 need a synchronization mechanism to correctly process the information. In the presented gate, the elaboration of information is based on a close-range interaction. If one of the two inputs anticipates the other at the gate the operation will not produce the correct result. The synchronization mechanism available for skyrmions are the following

- **Notch synchronization** - a notch is placed on one or both sides of the track in which the skyrmion moves. When the skyrmion reach the notch, it cannot proceed further in the line. The notch acts as a pinning center for the skyrmion. [21] Applying a current pulse the skyrmion is depinned from the notch and can proceed further as shown in figure 2.7.

- **VCMA synchronization** - VCMA effect can be used to create a local barrier for magnetization. In particular in thin films based on a ferromagnet-oxide interface, applying an electrical field across the interface can modulate locally the strength of the perpendicular magnetic anisotropy. The modulation can be both positive or negative. When the case reach a zone of increased anisotropy the skyrmion stops . When the voltage is removed the skyrmion is allowed to proceed the motion. In case of negative voltage applied the skyrmion is able to enter the zone with the decreased anisotropy but cannot proceed further as long as the the negative voltage is present. When the voltage is removed the skyrmion can continue its movement. [59]. The described mechanism is shown in figure 2.7

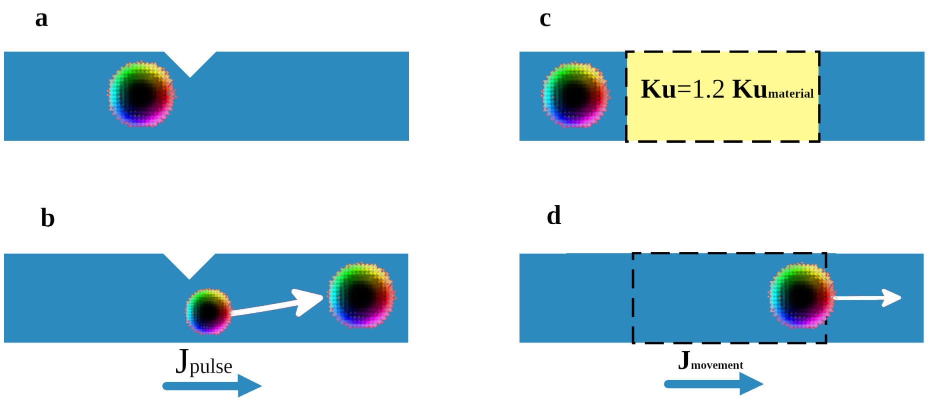


Figure 2.7: Synchronization methods for skyrmions. A notch in the track pins the skyrmion. A pulse unpins the skyrmion **(a-b)** A VCMA zone stops the motion of the skyrmion. When voltage is removed the skyrmion can proceed in the VCMA zone. **(c-d)**

## 2.3 Modeling of skyrmion devices in VHDL

To address the problem of the lack of a complete simulation environment in which heterogeneous system can be simulated, a possible solution that is proposed is to move the simulation environment completely in VHDL taking advantage of a selection of standard structures and their use limited to boolean logic purposes. The former aspect allows to develop a development environment based on standard blocks. Considering all the logic gates described in section 2.2 with an exception done for the ones based on skyrmion-domain wall technology for which other considerations should be done, some considerations useful for the modeling phase can be done:

- The magnetic textures involved, i.e skyrmions, are well localized in space. The size of the skyrmion might vary but limited to the confinement structures used. The logic result of the gates is linked to their position and movement inside the logic gates.
- The zone interested by the logic computation is localized on the texture of interest and small in space with respect to the complete gates
- The function realized by the logic is dependent only from a limited number of magnetic phenomena

The first point allows to describe the system not from the point of view of the single magnetic moments - the micromagnetic approach - but from the perspective of the collective magnetic textures used for the logic purposes. This description allows an enormous simplification of the calculations required to describe the system. This passage is not priceless because the collective description is by definition limited and restricted to some admitted functioning regimes. Moreover it requires that every collective phenomena of interest for the considered texture is described explicitly in the model or added separately. This approach is similar to the one employed with macrospin approximation for nanomagnets [60]. In this approach, a nanomagnet is approximated with a single domain magnetic particle - macrospin approximation - and its behavior is described with a single LLG equation. This approach is not suitable for the description of complex magnetic phenomena happening at a scale lower than the nanomagnet but when the interest is only in the collective movement of the magnetization, e.g. Nanomagnetic Logic section 8.3.2 , the description can give a valid mean to simplify computation. Regarding the second point the limited zone of interaction helps the compact description of the logic gate behavior. The limited interaction area allows to neglect the dependence between neighbor logic gates allowing the independent description of the single blocks describing a system. When the logic gate is evaluated there is no need to take into account the state of the other gates with exception of the input output relation between the gates. The last point allows to limit even further the description of the collective operation.

The phenomena linked with the collective description of the texture are indeed limited almost exclusively at the ones linked with the skyrmion drift velocity and skyrmion-skyrmion and skyrmion-edge interaction.

Starting from the above observations, a new design flow is proposed, based on a simplified modeling of a set of logic devices and basic components. This set of gates will allow the simulation of heterogeneous skyrmion-CMOS devices for the complete verification of complete logic.

### 2.3.1 Basic library

A basic set of gate was chosen to represent a logic system based on skyrmions. The chosen gate family for the demonstration of our proposal is the skyrmion logic family proposed by Chauwin et al. in 1.14. This family of gates is particularly suited for the design goal that was set, because the skyrmion is not modified in the elaboration and the gate structure is quite regular. The gates in this family are two: the AND/OR gate and the NOT/COPY gate. These gates, explained in section 2.2.2, takes advantage of the skyrmion-skyrmion and skyrmion-edge repulsion to realize boolean logic functions. In addition to these gates, to complete the description of a skyrmion logic system, other five blocks are considered: the basic track, a join of two tracks, a T structure, the cross structure and the duplication gate presented in section 2.2.1. The duplication gate is required by many logic functions in which information has to be distributed to multiple points in the circuit. An alternative to this gate is the NOT/COPY gate. The bottom and top tracks produce indeed a copy of the input information while in the central track its negated version is obtained. The other gates of this list allows to represent the different basic interconnections.

A variation of the basic track is also present to control the synchronization of gates, figure 2.8f. This track has an additional VCMA contact to stop the movement of the skyrmion moving in the track [61, 59, 62] All the presented gates constitutes our basic library for the realization of basic skyrmion circuits.

### 2.3.2 Gate basic structure

All the gates has been modeled in VHDL. The language has been chosen for the compatibility with common CMOS design flows. They share a common interface, shown in figure 2.9

The structure shown in figure represents the basic logic interface of the block with the control system and the other gates. The basic input/output function of the gates is realized by the skyrmion ports. As shown in figure the ports of the gate are bidirectional. All are able to emit or receive a skyrmion. The skyrmion in this system is represented by an event that is produced or received at the ports when the texture cross the point dividing the gates. The stimulus for the movement in

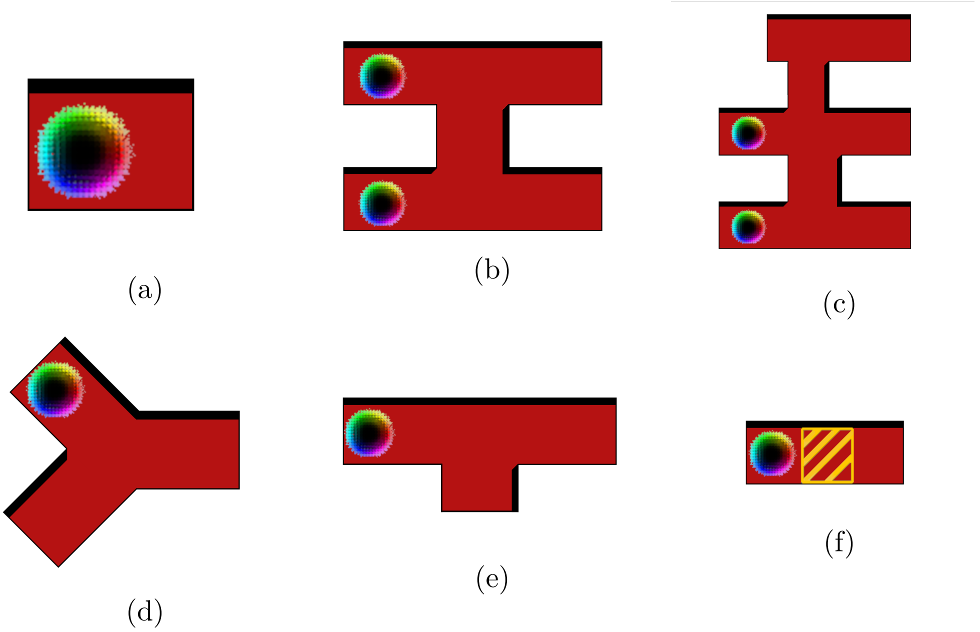


Figure 2.8: Basic structures for library. a) Line; b) AND/OR gate; c) NOT/COPY gate; d) JOIN gate; e) T structure; f) VCMA sync gate

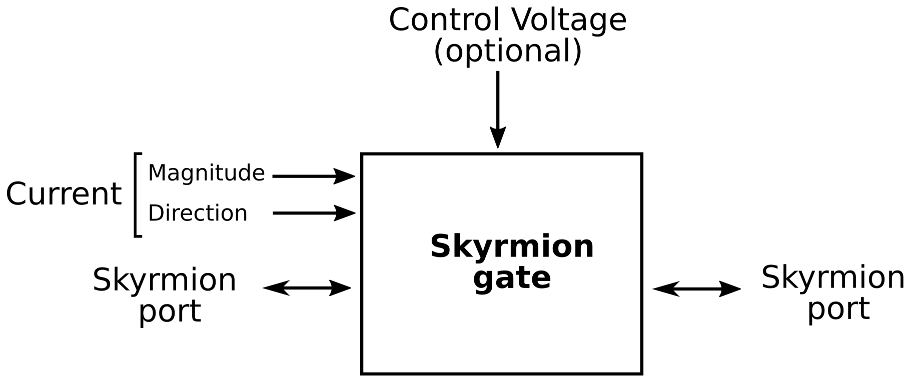


Figure 2.9: Skyrmion gates interface

the gate is controlled with the current ports. In particular the current stimulus is described by two inputs. One is used to give the magnitude of the current density crossing the port. The other is used to give to the gate the direction of current. Finally some gates are equipped with an additional input port used to control possible VCMA contacts present in the gate.

### 2.3.3 Toy model

The presented modeling for the different skyrmion gates is a simplified model of the movement of the skyrmions. Despite the clear limit of this modeling the description allows already a good approximation of the gate behavior useful for the simulation and verification purposes of this description.

The model implemented treats the skyrmion as a rigid structure in which the skyrmion is represented by the coordinates of its center. The position of the skyrmion is evolved following a non-inertial movement. In fact as shown in literature [8] a skyrmion in a ferromagnet when subjected to a stimulus have the tendency to move in the direction of the border and reduce its size until the equilibrium between the current stimulus and the border is reached. After that point the skyrmion moves with a steady state velocity in the line direction dependent from the equilibrium size of the skyrmion. In the case of the presented model, the skyrmion in presence of a current, that is the only stimulus admitted in the actual implementation, reaches immediately its steady state velocity and moves accordingly without considering the gradual increment of the speed given by the gradual decrease of the size and from the repulsion from the confinement border. The velocity of the skyrmion is obtained from the current magnitude and direction given to the skyrmion and the confinement. The difference between the structures is indeed represented by the shape of the confinement. The description of the confinement is done dividing the gate in different zones. The approximation of the confinement throughout the gate is realized by deleting the velocity components not admitted in that zone. This means that in the gate, as soon as the skyrmion reaches the aperture, zone number 6 in figure, the skyrmion vertical speed is enabled and the skyrmion position is updated considering also the vertical movement. In the same way when the skyrmion is in a zone with lateral confinement only the vertical movement is allowed. An example of the effect of confinement is shown in figure 2.10

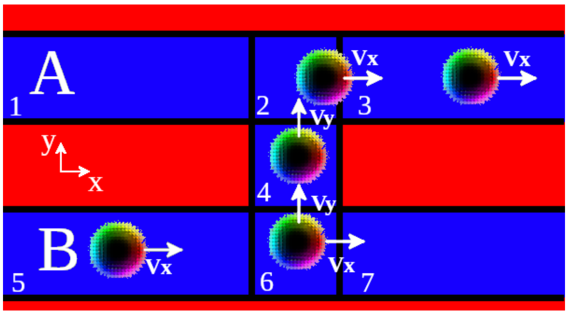


Figure 2.10: Gate partition for confinement effect approximation.

The gate modeled in VHDL has a series of procedures to manage the input/output of the gate and to update the position of the skyrmions traveling through it.

The block scheme of a single gate is shown in figure 2.11. The basic functionality is here explained. When a skyrmion translates from a gate to the other the skyrmion port receives a signal meaning that the gate connected at that port has emitted a skyrmion. The skyrmion receiver block now signals at the main block - Update skyrmion position - that a new skyrmion is present in the structure. This signal allows to the update position block to allocate a new skyrmion variable that from now on will keep track of the skyrmion position in the gate. The position of the skyrmion is then initialized depending on the port which signaled the skyrmion. Now, at every new time step, the position of the skyrmions in the structure will be updated. The position will be updated with the following equation:

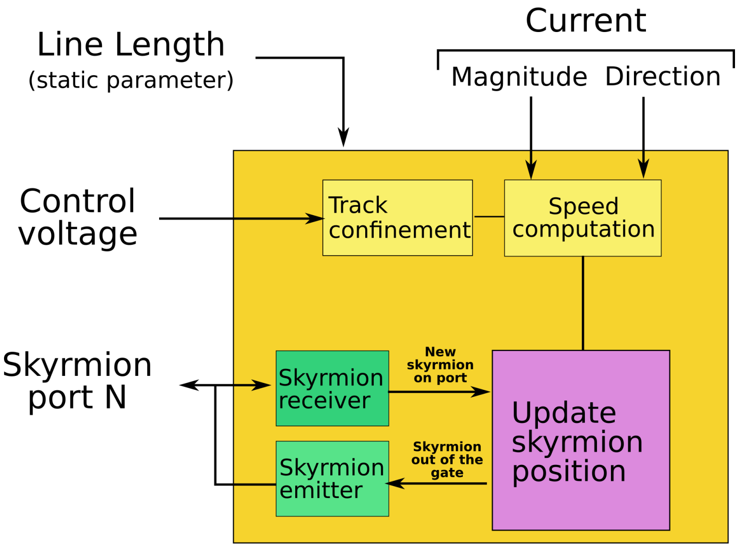


Figure 2.11: Logical structure of the modeled gate.

$$\boldsymbol{s_{t+1}} = \boldsymbol{s_t} + \boldsymbol{v} * t \tag{2.1}$$

where the $\boldsymbol{x_t}$ is the current position and $\boldsymbol{v_{skyrmion}}$ is the skyrmion speed along x and y directions. The speed is a function of both the position and the current stimulus. In particular, the model mimic the behavior of the skyrmion extracted from the simulations for a gate with the parameters reported in table 2.2. The speed for values from $1 \times 10^{10}\,\mathrm{A\,m^{-2}}$ to $10 \times 10^{10}\,\mathrm{A\,m^{-2}}$ were extracted from the simulations with a step of $1 \times 10^{10}\,\mathrm{A\,m^{-2}}$. For the missing current values, the speed is extracted as a linear fit between the closest values obtained in the simulation of curbed The speed values are provided by the speed computation block which computes the current speed of the skyrmion using as inputs the current in magnitude and direction and the confinement effects of the skyrmion. In particular this last parameter is computed by the track confinement block that signals what part of the gate the skyrmion is traveling through and what VCMA gates are active or

Table 2.2: Parameters used for micromagnetic simulations used as base for modeling

| SIMULATION PARAMETERS | | |
|---|---|---|
| Saturation Magnetization [52] | 1e6 | $\mathrm{A\,m^{-1}}$ |
| Uniaxial Anisotropy Constant [52] | 8e5 | $\mathrm{J\,m^{-2}}$ |
| Exchange Stiffness [52] | 2e-11 | |
| Damping constant [52] | 0.015 | |
| Spin Hall Angle [63] | 0.4 | |

not. From these inputs and knowing the current crossing the gate, the speed of the skyrmion is obtained. The position of the skyrmion continues to be incremented or decremented, depending on the current direction, until one of the coordinates is out of the defined area of the gate. At this point the update position block signals to the skyrmion emitter and then emits a signal to the correspondent skyrmion port.

### 2.3.4 Skyrmion conservative logic simulation example

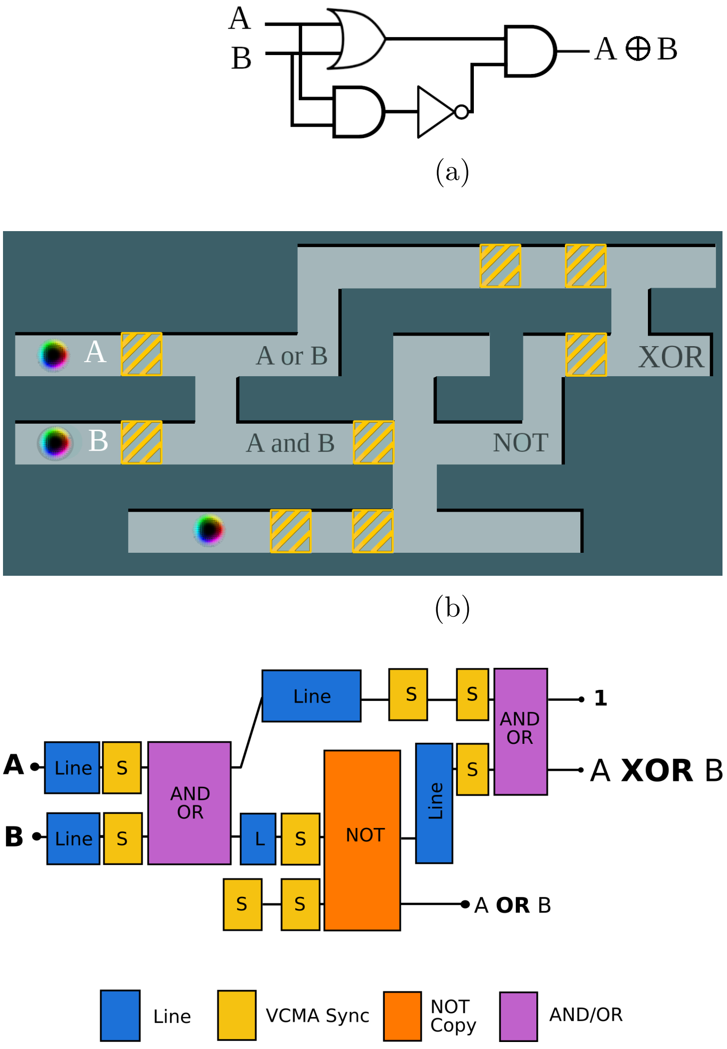
Now an example of the modeling and simulation of a gate is shown starting from basic XOR structure shown in figure 2.12. The structure is realized starting from the AND/OR and NOT/COPY structure. The XOR operation is obtained by the following logical relation $AxorB = (A + B) \cdot (A \cdot B)$ and the logic schema is given in figure 2.12a

The gate realize a XOR operation and will be used later in chapter 6 function in a proposal for an AES LIM implementation.

The basic functionality of the gate is here explained: the skyrmion are pushed forward in the direction of the AND/OR gate. Here the top track will produce the equivalent of an OR operation between the input operands. The bottom track will produce the equivalent of an AND operation. The top result is sent directly to the last section of the gate. The bottom result is now negated by means of the NOT/COPY gate on the bottom. The result reaches then the last gate, together with the result of the previous OR operation . The result of the complete XOR operation will be produced at the bottom output of the last gate. The yellow boxes in the gate shown represents synchronization elements built with with VCMA gates.

As shown the gate that is realized in skyrmion logic, is mapped directly with the blocks presented in the previous section. In particular for the mapping of this gate the current is distributed in the same way to all the blocks with the same direction. The connection scheme is shown in figure 2.12c.

In addition to the blocks shown in figure other three blocks were added to the testbench, a current generator, that is connected to all the blocks in the simulation. It sets the current stimulus for all the block. A synchronization control, that controls the behavior of the synchronization blocks. Two inputs blocks are then present to generate the skyrmion signals at the A and B ports.



(a)

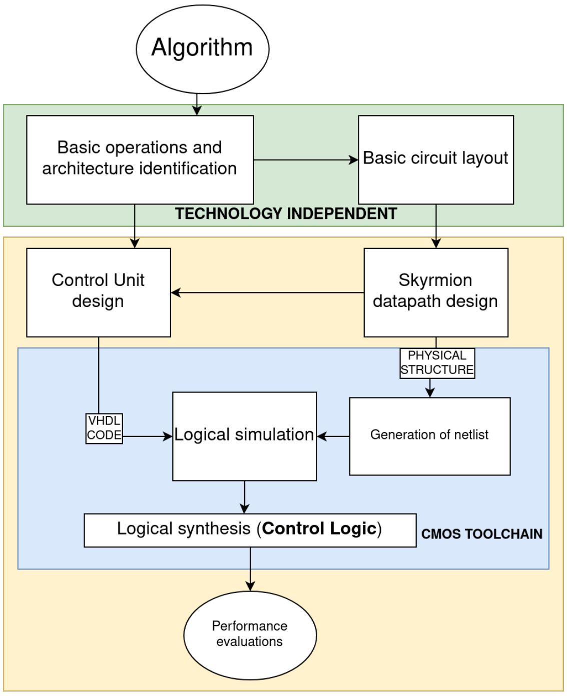
(b)

(c)

Figure 2.12: Mapping of XOR based on skyrmion logic with the skyrmion library

### 2.3.5 Design flow

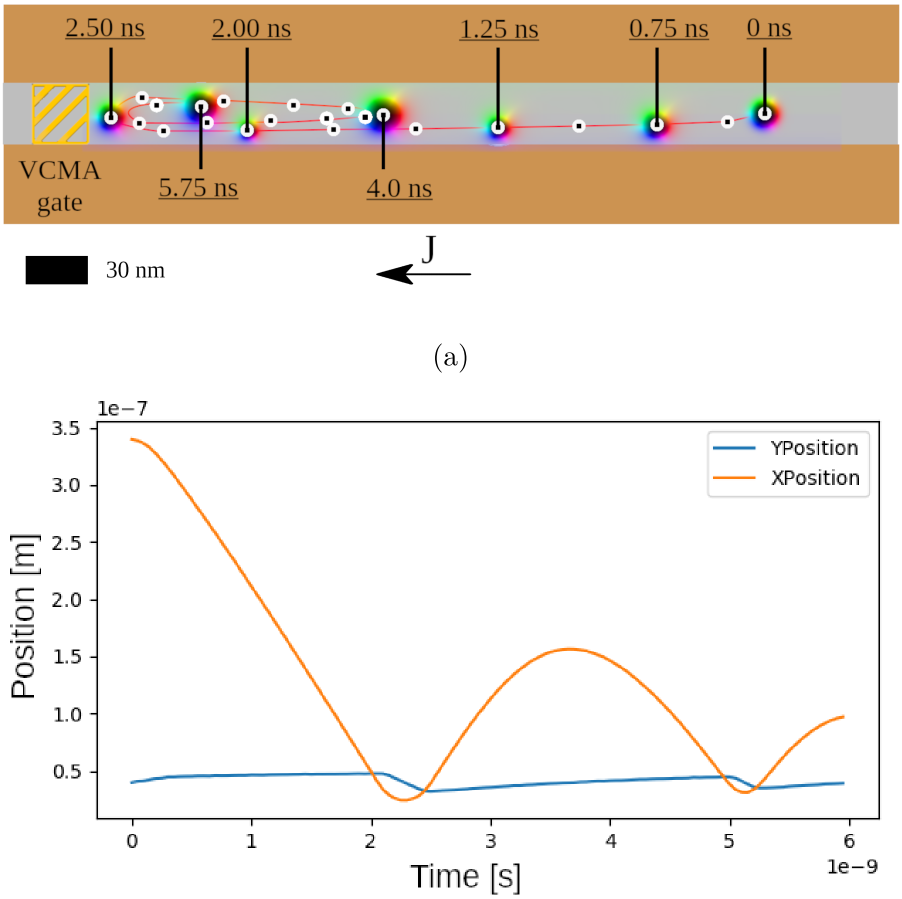
Starting from the above example a new design flow is possible for skyrmion based logic circuits. The technology independent part of the model is the same of a common design flow as showed in the previous section. The introduction of the model modifies the design in the phase in which the logic circuit design is mapped to the skyrmion technology. The presence of the skyrmion model allows, after the first design of the skyrmion datapath, to generate a netlist as the one shown in figure 2.12c and then simulate it together with the control unit in the same VHDL simulation environment. In this way all the control signals can be directly tested on the skyrmion gates to verify if the control is correct. At the



same time the complete functionality of the datapath can be tested. The times the simulation shows represents with an approximation the gate times of the system. For this reason the critical path found through this simulation can also serve as an approximation of the final critical path also to evaluate the overall working frequency. This is especially true for synchronous circuits as the one showed in the previous chapter. Nevertheless in case of asynchronous circuit or in case a strong optimization of the times is needed, an additional study of the critical components by means of micromagnetic simulation is still needed to understand what are the working limits of the structures. In the current modeling of the skyrmion movement, the breathing of the texture and the movements due to stabilization are not included in the model. This missing feature can force the user, in the aforementioned cases, to rely anyway on the physical simulations especially for systems with low damping and architectures that takes into account these effects.

A simple example of the last concept is shown in the following:

The system above is a curbed line realized with a W/CoFeB/MgO stack, the parameters are shown in table 2.2. The curb is 30 nm wide and 1 nm deep. The skyrmion is pushed with a current of $6 \times 10^{10}\,\mathrm{A\,m^{-2}}$ after an initial pulse of $20 \times 10^{10}\,\mathrm{A\,m^{-2}}$ applied for 0.3 ns to reduce the transition time from the relax state to the steady state. The movement of the skyrmion is stopped by a VCMA gate with anisotropy increased by 30%. As shown the skyrmion oscillates before stopping in front of the barrier. This effect can impact on the performance especially if a precise synchronization is needed. Especially in these cases the behavior depicted by the model is



(a)

(b)

Figure 2.13: Example of dynamical effect linked to skyrmion motion in systems with low damping. In a) the motion of the skyrmion in the track, in b) the evolution of the position of the center in time. The example is referred to a W/CoFeB/MgO with a damping constant of 0.015 and a current of $6 \times 10^{10}\,\mathrm{A\,m^{-2}}$. Due to the reduced damping, the skyrmion changes direction after the collision with the VCMA gates and bounces back.

not precise in time with respect to the real case. Nevertheless the presented model is useful even in this case as a tool of formal verification of the logic of the hybrid CMOS-Skyrmion system and gives a rough estimation of the critical components of the gate.

### 2.3.6 Conclusion and future prospects

In this chapter, a design flow based on a VHDL model of skyrmion logic gates was presented for the simulation of hybrid CMOS-Skyrmion systems. In particular, a new design methodology was proposed for logic circuits based on skyrmion logic circuits in order to allow the verification of the complete system. With the proposed modeling, the control circuitry realized in CMOS and the skyrmion logic gates can be simulated in the same simulation environment. Following this method the complete system can be evaluated at one time. In addition, information about the timing can be obtained using the model. It is possible to evaluate the critical path position and with an acceptable approximation its value. The model is limited by the usage of mean velocities for the gate modeling. In systems with low damping the information about the timing can be inaccurate due to the oscillation of skyrmion during its movement. At the same way in systems that takes advantage of dynamical effects like breathing modes or transient modes between stable states, the modeling is not accurate enough to give the correct information about the behavior of the system. A future advancement of the proposed approach would be to incorporate an analytical model for the description of the system and of the confinement potentials to mimic better the behavior of a real skyrmion system.

# Part I

# Logic in memory devices with skyrmions

The following part will go through the exploration of logic in memory devices based on skyrmion technology. Despite the goal of proposing new devices to solve known problems implementing specific algorithms in the target technology, the current part of the thesis has the goal of selecting case studies to better understand what are the capabilities and drawbacks of the use of skyrmions in complex architectures. The choice of exploring logic-in-memory devices was done for the reason that, in addition to the great promise for the use of skyrmions for memories, the additional logic capabilities, demonstrated in simulations suggests can be used to build more complex architectures. In particular, as will be shown in the next chapters, a direct communication from the memory to the logic gates is trivial in skyrmion systems. This capability allows to take advantage of information in memory without the need of reading operations that would limit inevitably its capabilities. For the aforementioned reasons, the LIM approach has been explored with skyrmion technology. The exploration presented in this thesis follows a gradual approach in the application of the LIM concept. The architecture presented are increasingly more complex, especially in the interaction between the different cells and the overall control needed to realize the required function.

The exploration started from Content Addressable Memories. This kind of architecture is the more traditional with respect to the one proposed in the thesis. In this algorithm every cell computes independently its result and the final result is finally obtained putting all together the results of the single operations. In addition, as will be better explained in the next chapter, the application of a search algorithms in a non-volatile memory that is likely to be employed at the level of the main memory, can open interesting scenarios, reducing the movement of information along the memory hierarchy. CAM memories are almost exclusively used at the top of the memory hierarchy.

Next, a LIM architecture for Maximum and minimum search is explored. This architecture differently from the previous executes a proper elaboration of information. In addition the algorithm is executed in many iterations of the basic cycle requiring a proper control of the information. The realization of this algorithm allows a direct comparison with a LIM implementation in CMOS for the same algorithm that allows to understand what are the limitation of a small elaboration realized with different technologies.

The last explored scenario is a LIM device for AES cryptography. This last explored algorithm is much more complex and requires, as will be better explained later, a more complex architecture with respect to the previous examples. Cryptographic algorithms are based on matrix operations and elements shuffling. These operations require a complex communication mechanism between cells and a precise control of the information flow inside the memory. In addition as in the case of CAM memories the application of this algorithm in real system can be of interest for encryption of big amount of data thanks to the parallelism such implementation can offer. All these architectures will finally show a possible strategy to implement logic in memory with skyrmions and what are the main design constraints linked to the implementation of logic in this technology.

# Chapter 3

# Content addressable memories with skyrmions

In the framework of the present thesis the exploration of the LIM paradigm with skyrmions will start from the presentation of an implementation of a Content Addressable Memory (CAM) memory. This particular kind of memory represents in fact a first step in the direction of a memory with complex logic capabilities. CAM memory in fact has only a limited logic implemented and its function does not require a complex control. In addition there is no interaction between the different cells. This first example will allow to show clearly the capabilities of skyrmion logic and the characteristic of a logic design in this particular technology.

## 3.1 Content Addressable Memories

CAMs are a particular class of memories that allows to look for a particular content inside the memory without knowing in advance its position. CAM memories in fact usually take as input a word of the same length of the ones stored and give as output its address inside the memory. This kind of application is particularly interesting for fields like telecommunications [64, 65, 66] or data-intensive applications [67, 68]. In addition to its search function, CAM memory behaves also as common Random Access Memory (RAM). The typical structure of a CAM memory is the one showed in figure 3.1. The search input is fed to the array by means of the search line (SL) signal. Every bit of the search line signal is then sent to the corresponding memory cells inside the array.

CAM memory cells can be divided in two categories basing on the value that Search Line (SL) bits can take: if the SL bits can take only value 1 and 0 the memory is called Binary CAM; in case SL bits can take 0,1 and X or "don't care" values the device is called ternary CAM. The "don't care" signal produces a match both in case the memory cell contains a 1 or a 0. If the search function is active every cell will

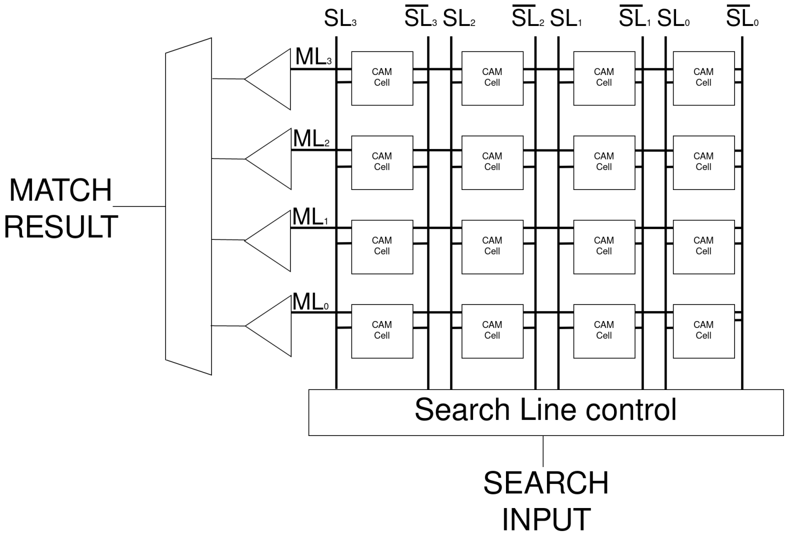


Figure 3.1: CAM memory basic schematic

produce a match only in case the bit contained in the cell is equal to the bit of SL. The match information is used at this point in CMOS architectures to pull down a line connected to a sense amplifier that will produce the match signal for the single word. Similar behavior is used for other technologies where the memory element substitutes the transistor memory cell and is used in combination with regular transistor to produce a similar effect on a match line [69, 70, 71]. Regarding LIM architectures CAM cell is an important building block. It represents a basic form of elaboration that can be used as a seed to build more complex logic. In the context of this thesis will give an initial point of view of the challenges of logic integration of in-memory elaboration.

## 3.2 CAM devices with skyrmion technology

CAM devices are particularly attracting as first devices for LIM architectures for their limited complexity, which allows to better focus on the optimization of the proposed structure. In addition, being the skyrmion a technology particularly suited for memories [21, 52], this additional functionality can be really beneficial to skymion memory devices and even more importantly to the memory hierarchy. The benefit for the memory device is that, as it will be clear later, the additional cell comes with a limited cost especially in terms of area. The benefit for the hierarchy comes from the fact that, being able to execute search operations closer to the bottom of the memory hierarchy, it is possible to reduce with respect to the actual systems, the miss occurrences in faster memories of higher level. In a complete hierarchy slower non-volatile memories are at the bottom where usually they are non-volatile devices that provide high density and low cost per bit. Going up in the hierarchy it is possible to find faster and faster memories where the access times are increasingly smaller and implementations costs rise. By construction, CAM memories are employed almost always in the faster memories like SRAMs. In literature, also proposals for DRAMs based CAM [72] are present. This means that if some information needs to be processed, it should be fetched from the bottom of the hierarchy and brought up to the CAM equipped memory and then filtered. In this scenario, being the transfer of information a cost in terms of time and energy, executing operations down in the hierarchy can lead to great benefits in terms of efficiency of operations and overall power consumption. This is especially interesting in the scenario in which the non-volatile memory can have an high density like skyrmion racetracks.

### 3.2.1 CAM Cell design

#### Memory track

The CAM Cell proposed is based on a skyrmion racetrack memory. A section of the memory is shown in figure 3.2. The curb in the ferromagnet is designed to give to the skyrmion an additional protection from annihilation. This protection increase the threshold current over which the skyrmion is annihilated at the borders. In addition as reported by [73], the skyrmion moving in the constriction exhibit an increased speed with respect to a plain nanotrack. Similar confinement effect can be obtained with locally engineered anisotropy values as proposed in [74]. The track chosen for the implementation has a curb 30 nm wide and with a depth of 1 nm. The reference stack used for the simulation is $\mathrm{W\backslash Co_{20}Fe_{60}B_{20}\backslash MgO}$. This stack has been chosen for the strong anisotropy and the high spin hall angle [63]. The parameters of the reported stack are shown in table 3.1

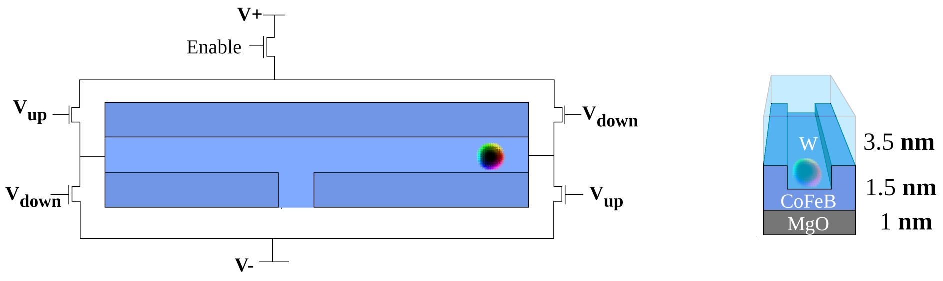


Figure 3.2: Reference stack for skyrmion movement. The curb confine the movement of the skyrmion to enhance its stability at high currents

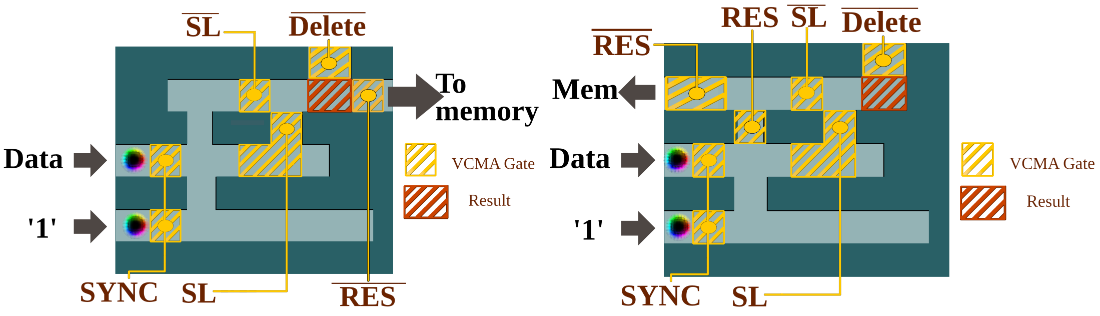
The binary information is encoded in the track with presence of skyrmion. The bits are stored in the memory in defined portions of the nanotrack. These portion will be considered the basic memory cells used to implement the CAM functionality.

Table 3.1: Parameters used for micromagnetic simulations and current distribution computation

| SIMULATION PARAMETERS | | |
|---|---|---|
| Saturation Magnetization [52] | 1e6 | $\mathrm{A\,m^{-1}}$ |
| Uniaxial Anisotropy Constant [52] | 8e5 | $\mathrm{J\,m^{-2}}$ |
| Exchange Stiffness [52] | 2e-11 | |
| Damping constant [75] | 0.026 | |
| Spin Hall Angle [63] | 0.4 | |
| Film resistivity [63] | 165 | µΩ cm |

### 3.2.2 Skyrmion CAM

A possible solution for implementing a CAM cell with skyrmion memories is represented by the design showed in figure 3.3.

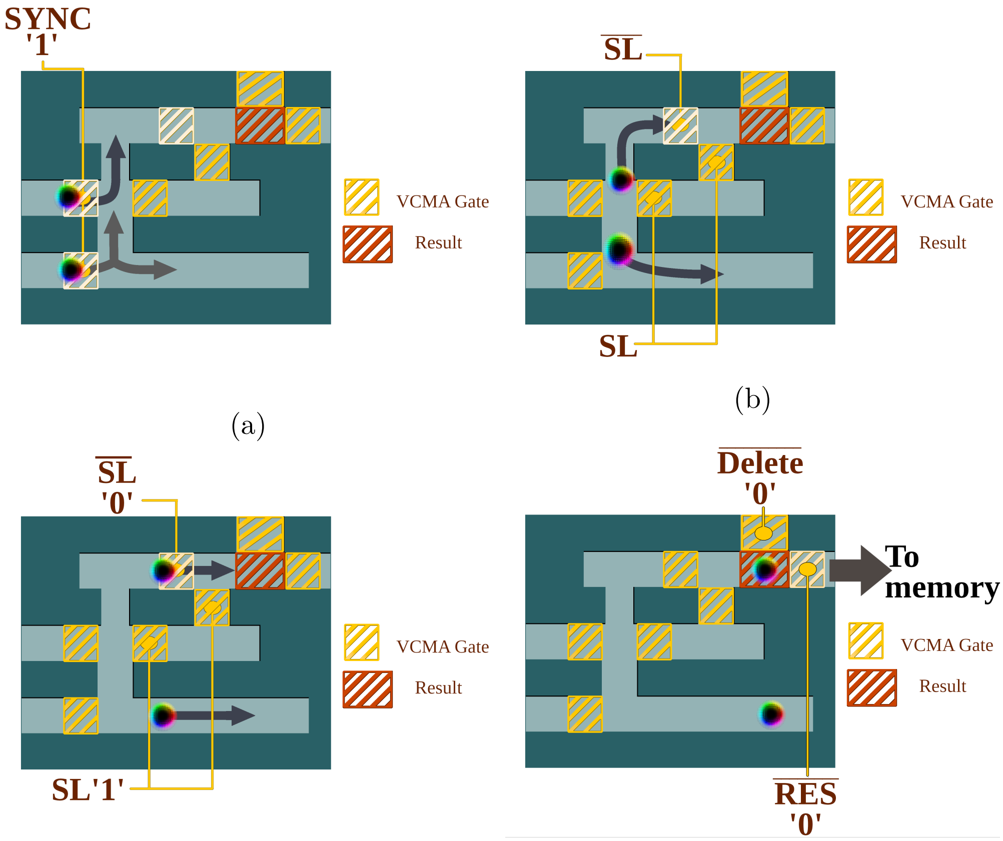


(a)

(b)

Figure 3.3: Skyrmion CAM Cell. In a) the cell with forward restore path to memory. In b) the restore path is in the memory direction

The presented gate takes advantage of the structure of a NOT gate proposed by Chauwin et al. [55] with additional VCMA filtering option to realize the matching function. This function is logically equivalent to a XNOR function. The cell, indeed, signals a match only in case the SL and the data contained into the cell are equal. The cell elaborates the information encoded in the presence of skyrmion at the input of the gate and after guiding the information through the patterned structure produces the match result in the result zone of the gate. Here, the match result, in form of skyrmion, can be read or reused for further computation. The $SL$ signal and its negated version $\overline{SL}$ are used to control two VCMA gates. The signals are used to change locally the anisotropy, allowing the moving skyrmion to pass in one case or stopping its movement in front of the VCMA gate. Finally, an auxiliary skyrmion is placed at the bottom track to allow the correct operation of the complete gate. In addition to the main gate inputs, two additional sets of controls are present in the cell. The first is placed at the entrance of the gate, guided by the signal SYNC, where a set of VCMA gates is used to synchronize the whole operation. The gate to operate needs a synchronized movement of the auxiliary and the data skyrmion. The second is placed in the top section of the gate, guided by the signals RES and DELETE. Here, this set of VCMA gates is used to keep the result skyrmion during evaluation and then, in a subsequent phase, it is used to selectively guide the initial information of the gate back to the memory. These controls are used also to erase possible additional skyrmions left into the gate before the next operation can take place. Both these operations, delete and restore, are necessary to correctly restore information inside the memory after a search operation, as will be discussed in section 3.2.3. The cell functionality is shown in figure 3.4



(a)

(b)

(c)

(d)

Figure 3.4: Example of CAM Cell operation. In the example data input is 1 and SL is 1. (a)The operation is triggered. (b)The skyrmions continue their movement based on the mutual repulsion. (c)The search signal SL is equal to ’1’, the skyrmion is allowed to cross the VCMA gate. (d)The result is ready in the evaluation region. The result can be deleted or sent to memory

The functionality can be virtually divided in 3 stages. In the first stage, represented in figure 3.4a, the VCMA gates used for synchronization are opened, removing the voltage and the information can proceed inside the structure. The central track is the one connected with the main memory and carries the data. The bottom track as shown in figure 3.3, receives always a skyrmion. This auxiliary skyrmion is indeed necessary for the functionality of the gate. Then, if the data track has a skyrmion, it will end up in the top track. At the same time, the auxiliary skyrmion will continue its movement in the bottom track, leaving the central one empty, as shown in figure 3.4b. These movement as explained in section 2.2.2, is the result of the joint effect of skyrmion-skyrmion and skyrmion-edge repulsion effect. In case a skyrmion in the data track is not present, the auxiliary skyrmion will continue in the central track, not being repelled by any other skyrmion.

In the second stage, the information is filtered on the base of the SL signal. The skyrmion in the upper track reach the VCMA gates guided by the SL signal and its negated version. Now, if the SL signal is 1, the VCMA gate on top is open, leaving the skyrmion in the top track pass through as shown in figure 3.4c. At the same time, the gate in the central track, guided is closed. On the contrary, in case SL signal is 0 only the skyrmions traveling in the central track will pass through, while the gate in the top track will be closed. Both the paths are then joined to reach the final evaluation zone. In the third and last stage, figure 3.4d, finally, the result is ready in the result zone, indicated with a red square in figure. In case a match was obtained, a skyrmion is present at the output, otherwise the result zone is empty. The success of the match operation is equivalent to the situation in which both the information coming from the data input and the SL signal are equal.

After the evaluation two possible operations can be performed: the skyrmion at the output of the gate can be used for further elaborations cascading other skyrmion logic gates or the original information can be restored by means of DELETE and RESTORE signals.

### 3.2.3 Input bit restore

An important feature of the proposed gate is the one allowing to the information, given as input, to be restored after the match operation is concluded. This is fundamental for a successful application in memories, where the stored data should not be lost in elaboration. To execute a correct restore operation, the gate exploits the knowledge of the SL signal that produced the match result The first part of the gate, the NOT/COPY operation produces on the central track a logic NOT of the input, and on the top track an identity function with respect to the input information. Starting from this information, depending on the value SL had in the search operation, the original information can be blocked at the VCMA gate guided by $\overline{SL}$ in the top track or at the end of the gate.

Then it is possible to define 2 different cases depending on the last SL value:

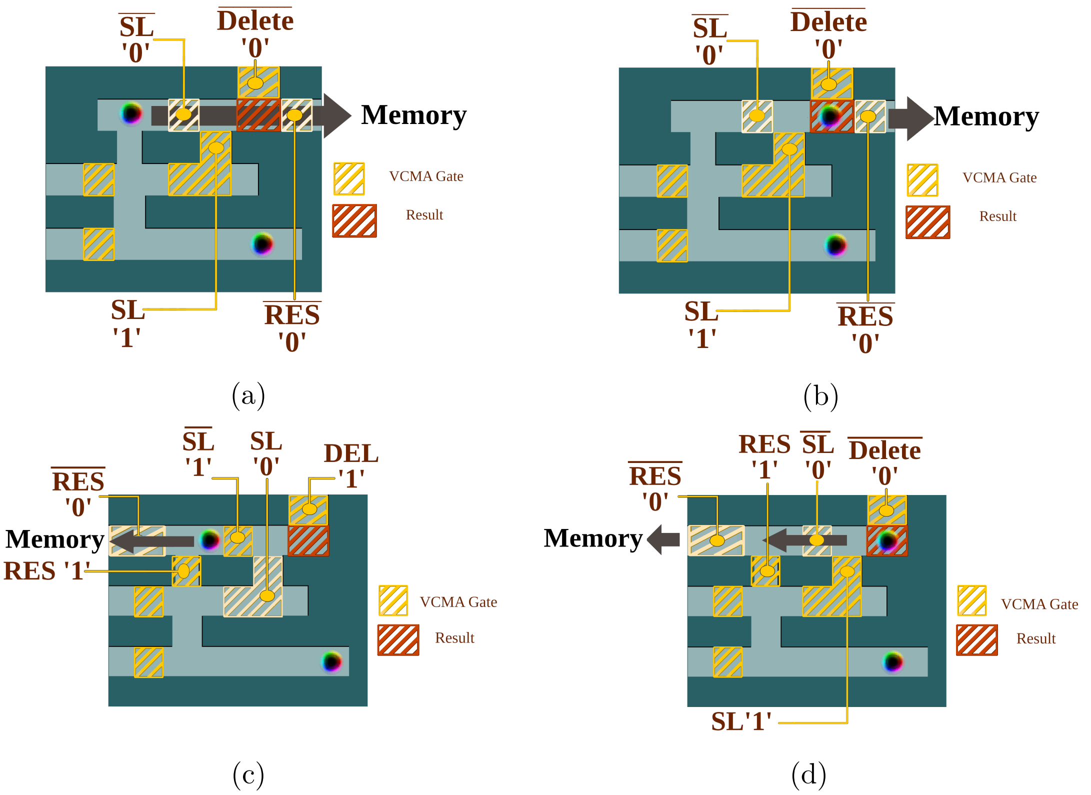
- **SL = 1** the information in the result zone is the input information.
- **SL = 0** the information is blocked behind the VCMA gate controlled by SL and the eventual skyrmion in the result zone is the auxiliary skyrmion.

As shown in figure 3.3. Two versions of the gate have been designed to allow the restore operation in forward direction, figure 3.3a and in the backward direction, figure 3.3b. In the first design when the SL signal was equal to 1, the information, in the result zone has to be pushed forward and the restore operation is complete. The $\overline{RES}$ signal is then set to 1, lowering the barrier at the output, the information can proceed out of the gate, figure 3.5b. It is important to note that if the input bit was zero, no skyrmion will be sent out of the gate, allowing the correct restore of the value zero in memory. In case the SL signal was equal to 0, figure 3.5a the restore operation requires, at first, the clear of the eventual skyrmion in the result zone to avoid errors in the restore operation. The delete signal is then set to 1 and the top gate is open. A current pulse, needed for information movement will then push away the skyrmion. The SL signal and the RES signal are then set to 1 to open both the gates in the top track. The information can leave the gate in the same way as the previous case.

The second design differently from the first solution, is able to restore the initial information on the same side where the input came from. In this design, in the case in which SL signal was equal to 1 during the match operation, the skyrmion has to travel back through the whole gate to reach the exit on the input side, figure 3.5d. The RES signal is then set to 1 and the skyrmion, pushed by a reversed current, can travel back to memory. In case the SL signal was 0 in the match operation, the skyrmion is already close to the exit. In this situation, the RES signal is activated and the skyrmion pushed back into memory, as shown in figure 3.5c The two different restore procedures are shown in figure 3.5

### 3.2.4 Methods

The cell has been simulated in the micromagnetic framework by means of Mumax3 [19, 53] in order to verify its function and performance. The parameters used in the simulation are reported in table 3.1. The damping parameter used in simulation was chosen higher than the one reported in [52] to reduce the dynamic effects linked to the low damping, like the oscillations at the VCMA gates discussed in chapter 2.3.5. The value chosen for the simulation was taken from [75], in which the authors experimentally measured the damping value of a W/CoFeB/MgO stack. Higher values of the damping factor would have allowed to further reduce the oscillations at the VCMA gates but for the chosen stack the values would have been too far from the ones reported in literature. To appreciate this reduced oscillation at the VCMA gates it is suggested to refer to [62]. In this paper the authors in the supplementary material show how in a Co/Pt stack, that has a damping usually in



(a)

(b)

(c)

(d)

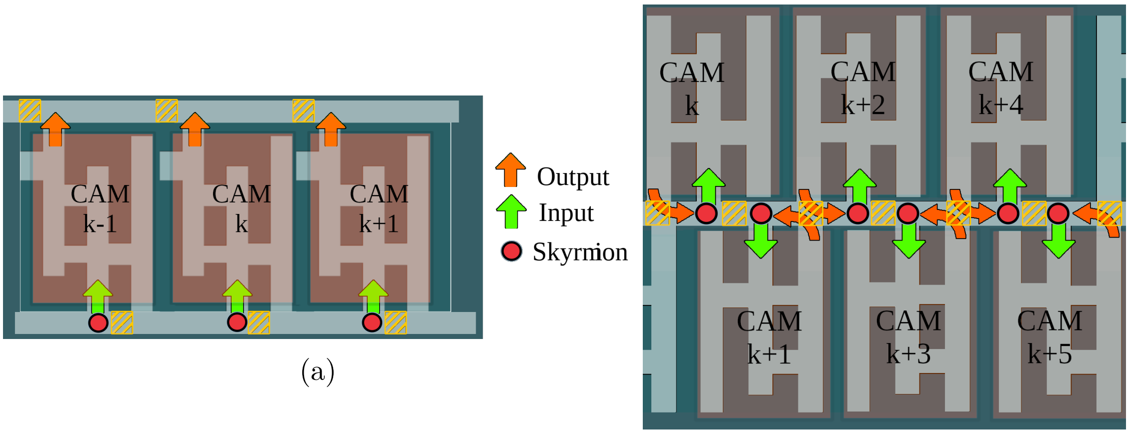
Figure 3.5: Restore procedure for the CAM cell with forward restore (a-b) and the CAM cell with backward restore mechanism (c-d)

the order of 0.1, the skyrmions reaching the gate stop exactly in front of the gate without oscillations.

The skyrmion confinement is realized with the curbed structure showed in section 3.2.1.

### 3.2.5 Integration in memories

Being a memory application, the integration with standard racetrack memories is very important for the proposed design. As shown in the previous section, the use of such device is especially beneficial if it is integrated inside a skyrmion memory, where the information already in a skyrmion state can be easily routed into the CAM cell. As a standalone device, other technologies are at the current state of development more beneficial especially because with similar values of energy consumption, they are able to produce a result with a superior speed, that in the proposed device is limited by the skyrmion expulsion current threshold. In this framework, the proposed skyrmion device has been designed with two possible restoring mechanisms as shown in figure 3.3. With these two designs the integration



(a)

(b)

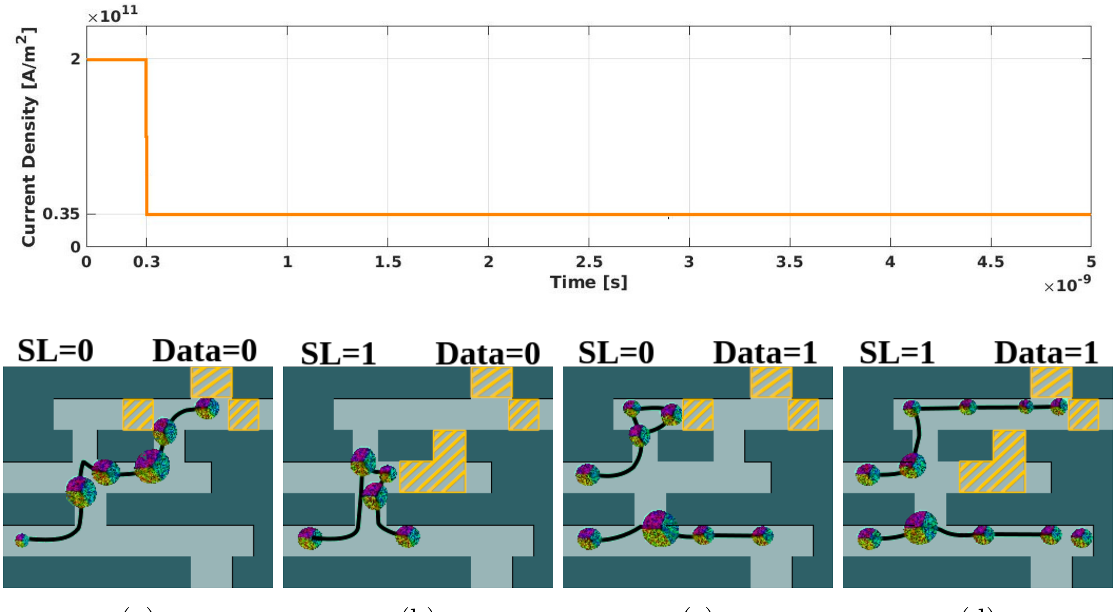
Figure 3.6: Possible connection schemes for CAM cells to the main memory. a) CAM cell connections on a single side. In this case, the restore mechanism sends the information on another skyrmion memory track. b) CAM cell connections on both sides.

has two possible topologies. As shown in figure 3.6a, the cells can be connected on one side of the memory. In this kind of configuration, after the match operation the information is immediately restored in memory in another section of the track. This configuration allows a smaller lateral integration but imposes a lower linear density to the memory. The minimum distance between bit cells is in fact the distance between two consecutive input tracks of the CAM elaboration. With this first topology the minimum distance between two consecutive bit cell is 210 nm. The minimum stable distance between skyrmions was obtained from simulations of a skyrmion memory track with the same characteristics of the gate. This minimum distance is 60 nm. The linear density of information is thus reduced of a factor of 3.5.

To further increase the linear density, the second topology can be adopted, positioning CAM cells on both sides of the memory track. This topology requires the restoring mechanism to send information back to the same side where the information was coming from as shown in figure 3.3b to not lose the original information. The information with this restoring mechanism is restored on the same side of the input. The density in this case is higher allowing to place consecutive skyrmions at a minimum distance of 90 nm, only 1.5 times higher than a plain nanotrack. The cost of implementation of this second topology is the control design. The restore operations cannot be completed on both sides at the same time. To avoid collisions between neighbor skyrmions. In addition the design of the contacts required for current generation is more complex, currents are needed in both directions, starting from the memory track and the restore operations from the two sides of the racetrack cannot be performed at the same time . In the first topology the current control is simpler: the current needed, in this case, is in a single direction and the restore operations can be performed all at the same time.

### 3.2.6 Micromagnetic simulations and performance evaluation

The micromagnetic simulation showing the complete functionality of the cell are showed in figure 3.7. As can be seen in figure the current needed to move the skyrmion through the structures is composed by a short pulse to start the operation and a lower constant current that, in sequence, produce the movement. In the case shown in figure the value of the current density for the pulse was chosen at a value of $20 \times 10^{10}\,\mathrm{A\,m^{-2}}$. The constant current used then to continue the movement had a density ranging from $4 \times 10^{10}\,\mathrm{A\,m^{-2}}$ to $8 \times 10^{10}\,\mathrm{A\,m^{-2}}$. The gate was tested under a constant positive bias field of 30 mT. This field was chosen to reduce the expansion of the skyrmion when pushed by an electrical current in the junctions. With a bias field the ringing of the skyrmion is reduced, leading to a more stable movement.

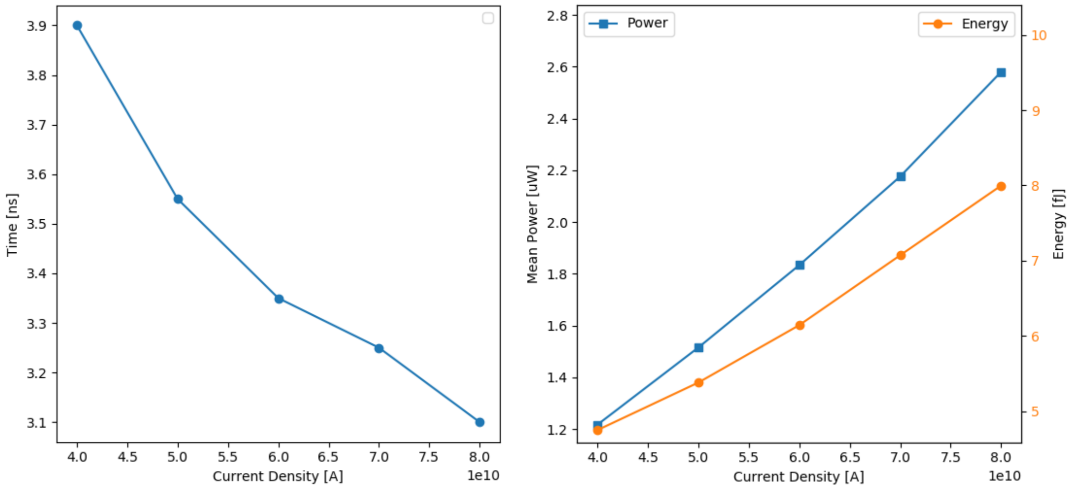


(a) (b) (c) (d)

Figure 3.7: Micromagnetic simulations of the skyrmion CAM cell. (a)Stored bit=0 SL=0. (b)Stored bit=0 SL=1. (c)Stored bit=1 SL=0. (d)Stored bit=1 SL=1.

In figure 3.7, all the combinations of SL and input bit are shown. As expected only the combinations with equal SL and input value registered a match, a skyrmion reached the result zone.

To evaluate the performance of the gate, the micromagnetic simulations were executed increasing the driving current up to the values in which the gate did not perform anymore the required function. The performance registered are reported in figure 3.8a.



(a) (b)

Figure 3.8: a)Skyrmion CAM search time and b)power and energy performance as functions of the driving current density. ·

As shown in figure, the gate cannot sustain current densities higher than $8 \times 10^{10}\,\mathrm{A\,m^{-2}}$ due to the expulsion of the skyrmion from the track and the wrong movement of the auxiliary skyrmion in the first junction. The current tested refers to a spin hall angle of 0.4, as presented in literature [63] for the chosen stack composition. With this current density the minimum timing to produce the result is equal to 3.1 ns. The longest path the skyrmion has to travel is correspondent to the case with both input and SL signal at 0, figure 3.7a. After the timing performance, also the power and energy were computed. The values were obtained starting from the sheet resistance of the reference W/CoFeB/MgO stack, that was considered equal to $165 \times 10^{-8}\,\Omega\,\mathrm{m}$ [63] neglecting the possible loss of efficiency due to eventual partitioning effects between the different metal layers of the stack. The power for the generation of the bias field has not been taken into account, it is assumed that, being constant, the bias is provided by a permanent source, like a permanent magnet, that will not require any power during circuit operation. The power was computed with the following equation:

$$P = \rho_{sheet} l_{gate} J^2 S_{gate} \tag{3.1}$$

where $\rho_{sheet}$ is the resistivity of the W/CoFeB/MgO stack, $l_{gate}$ is the length of the gate, $J$ is the current density applied and $S_{gate}$ is the section crossed by the injected current. An offset should be added to this value in order to take into account the cost of the nucleation of the auxiliary skyrmion present in the bottom track. Different techniques, some of which presented in section 1.2.5 can be applied to the nucleation of the auxiliary skyrmion. Skyrmion nucleation based on current is not suited for this kind of structure for two reasons: the space required for skyrmion stabilization would impact negatively on the gate density, the energy required for such operation would be high with respect to the circuit energy performances that, being employed in structures with high parallelism, can result in an unacceptable value for energy consumption of the complete circuit. A solution that is promising and suitable for the proposed CAM design is the one presented by [76], this solution require lower-power compared with other solutions and it is based on VCMA effect, already employed for synchronization and circuit functioning in the circle. As already presented in section 1.2.5, the reported power consumption depends only on the capacitance of the VCMA gate, that for the required dimensions, under $100\,\mathrm{nm}^2$ is expected to be lower than 6fJ. The minimum registered values for power consumption were registered at $4 \times 10^{10}\,\mathrm{A\,m^{-2}}$ and were equal to $1.21\,\mu\mathrm{W}$. The minimum energy consumption computed for the proposed design was 4.7 fJ. This current density is equal to an execution time of the search operation of 5 ns. Finally regarding the area, a value of $0.054\,\mu\mathrm{m}^2$ was computed.

The values obtained were compared with both high-speed and low-power CMOS implementations and the results are shown in figure 3.2. The performance of the CAM memories are expressed without taking into account the contribution of the read time of the circuit This because the read mechanism employed can be different (refer to section 1.2.6). In particular the most promising mechanism is the reading by means of magneto-tunnel junction. In this case the device during reading operation is equivalent to a varying resistance dependent on the presence of the skyrmion in its free layer. In this framework, the read operation can be used to obtain the final match value. A comprehensive study of the possible reading and matching solutions is out of the scope of this study. The dimension of the array used in table for comparison is 32x32 in order to be directly comparable with the studies presented in [77], [78], [79] and [80]. In the energy value presented in table also the write energy for the auxiliary skyrmion was considered that equals to 6fJ.

As shown in the table, the skyrmion implementation is worse compared to CMOS implementations for search speed and energy per search. At the same time, the skyrmion memory requires a smaller area compared to the SRAM implementation. This shows how this kind of solution can be more suitable in case of bigger devices, one of most important limitations of CAM is, in fact, the low density. Finally, the two technologies have an important difference given by the volatility of information. The skyrmion implementations allows to store information and maintain it without the need to give additional power to the circuit while the SRAM memory requires a continuous energy expense in order to not lose the information stored. In presence of physical synchronization the memory can be even completely

Table 3.2: Performance comparison between skyrmion based CAM and CMOS implementation - 32 x 32 CAM array

| | Search speed | Energy $[fJ]$ | Area $[um^2]$ | Non-volatile |
|---|---|---|---|---|
| CMOS - Mahendra, Mishra, and Dandapat [77] | 79 | 930 | ND | No |
| CMOS - Pagiamtzis and Sheikholeslami [78] | 1.18 | 2163 | 1536 | No |
| CMOS - Chang and Liao [79] | 0.4 | 8920 | ND | No |
| CMOS - Datta et al. [80] | 1.2 | 274 | 1597 | No |
| **Proposed Skyrmion - high speed** | 3.1 | 14336 | 0.054 | Yes |
| **Proposed Skyrmion - low power** | 3.9 | 10956 | 0.054 | Yes |

disconnected from the power supply while with the use of VCMA barriers, a mean to maintain the electrical field across the gate is still needed.

In addition to the previous comparisons the proposed architecture has been compared with different implementations of CAM Cells with MRAM technology.

Table 3.3: Performance comparison between skyrmion based CAM and other non-volatile CAM implementations

| | Search speed | Energy/bit $[fJ]$ | Area/bit $[um^2]$ |
|---|---|---|---|
| STT CAM-1 [81] | 0.2 | 37.37 | 6.84 |
| STT CAM-2 [82] | 1.28 | 5.07 | 8.28 |
| STT CAM-3 [83] | 0.17 | 0.17 | 10.76 |
| **Skyrmion Proposed** | 3.1 | 14 | 0.054 |

At the state of the art, even compared with other memory technologies, the skyrmion CAM implementation is worse in speed and power. The high power consumption comes from the dimensions of the gate and from the maximum speed achievable from the device. These performance comes directly from the resistance of the stack and from the efficiency of the charge-spin current conversion expressed by the spin hall angle equal to 0.4 in the chosen material stack. Another impacting factor on this study are the dimensions of the skyrmion. Maintaining similar movement speeds, the reduction of the dimensions of the skyrmions would allow to both reduce length and width of the proposed gate. The smaller dimensions would bring a reduction in search time and in the power required for the gate functioning. This would allow to obtain higher performances in both time and energy. At the same time, the implementation of such circuit shows a very high density compared with other implementations. This suggests that this implementation of skyrmion CAM memory is suited in cases in which the required CAM memory capacity is high.

## 3.3 Partial conclusions and improvements

In this chapter a CAM memory based on skyrmion was demonstrated, the proposed solution takes advantage of skyrmion edge and skyrmion mutual repulsion to realize a non-destructive search logic for skyrmions. Compared with state of the art volatile and non-volatile technologies, the design showed a good area occupation and consequently an high memory density while the performance in terms of energy and search time were worse compared to other technological implementations. This suggests that at the current state of the art the employment of such technology is useful only when an high density is required or when skyrmion technology is employed for information memorization, while for high performance or low power implementations other technologies remain preferable. In the presented study it was underlined how a skyrmion racetrack memory can be converted easily in a CAM memory without the need of expensive modifications to the basic racetrack and without reducing drastically the main memory linear density. This characteristic is very impacting on the read and write performance of the memory that depend on the racetrack length. The study presented investigated the performance of the circuit related to the skyrmion elaboration leaving aside the contribution of the surrounding circuitry. In particular the study of the reading and the match circuitry would be an important development to have a complete evaluation of a complete CAM device realized with this technology. Another interesting development of the presented study is the study of possible modifications to the material stack of the device in order to understand what are the maximum performance achievable from the presented implementation. An example is the tuning of the anisotropy and saturation magnetization values. Different combinations of these two values were reported in literature for the W/CoFeB/MgO stack, especially under ion irradiation. Same applies to damping and DMI constants. The exploration could allow to find the best material and stimuli conditions to bridge the gap between this and other implementations in terms of energetic and search performance.

# Chapter 4

# Final conclusions and prospects

In this thesis the problem of computation in memory with skyrmions was addressed. Starting from a methodology different implementations were explored and evaluated. The results from the performance evaluations suggests that skyrmion technology in the conditions explored still need some study and improvement for what concerns the energetic performance and further studies are required to optimize this characteristic. At the same time the skyrmion proved to be excellent in the density achievable from the proposed circuits. In all the proposed solution, but especially in the CAM and AES case the implementation proved to be able to pack a great number of gates in a small area. This aspect is particularly interesting in logic-in-memory implementations where the impact on the die area is usually high. In the skyrmion case this achievement is even more important. In this scenario the additional area occupied can have two negative effects: diminish the density of information that is one of the most important characteristic of skyrmion memories and slow down the read and write process of the memory. This process is based on shift operation and a lower density can represent a lower read and write speed making the memory more difficult to be integrated in real systems. For the future it is important to study further the problem of logic in memory with skyrmion. The characteristics underlined in this thesis shows a good potential for the current technology especially in systems in which the latency is not a critical parameter. In this direction it will be important to study both the possible optimizations applicable to the proposed systems and the material parameters that can further optimize the characteristics of the systems. Finally, regarding this first part it is important to underline that the destiny of such logic application is strictly linked to the success of the skyrmion memory. The device by themselves does not give enough advantages to be proposed as single logic devices. Similarly to what happened to bubble memory the main core of skyrmions remains still the memory and without the possibility to employ skyrmion primarily on memory, at the actual state of the art, the proposals of this thesis do not have a real reason to be used. This suggests that the research in this direction, logic-in-memory application, should focus on adapting itself to the best possible implementations for memory for which the proposed architectures should represent an expansion. In this direction the second proposal was done. The conjunction of PNML and skyrmion allows to connect to a good technology for memory a good technology for distributed logic. The generation of domains for skyrmion is a very natural process in magnetic materials usually employed for skyrmion devices. As shown in literature in [34] also the opposite is possible. This linked to the recent advance in PNML technology creates a very exciting scenario in which new kind of circuits are possible that depend less on electric control and can completely take advantage of the best characteristics of magnetic circuit. The presented study is a starting point and showed a simulative proof of the integration between PNML and skyrmion technology. It will be important in future to test this interface against real systems, designed for specific problem and understand if it's really worth it compared to other solutions. In addition to that it will be also interesting to understand if this same approach of communication between skyrmion and other magnetic technologies is usable for other computing paradigm that in the current moment are attracting great interest from a wide part of the electronic community.

# Chapter 5

# Minimum Maximum Search

The first algorithm explored for Logic In Memory (LIM) implementation with skyrmion is an algorithm for maximum and minimum search. The final circuit is designed to identify a maximum or a minimum among the words stored in memory in form of skyrmion. The device executes part of the algorithm in skyrmion technology inside the memory taking advantage of the elaboration unit to locally elaborate the stored information. The information is then read and compared with the results coming from the other elaborations to determine the maximum or the minimum. The presented algorithm was previously implemented by Vacca et al. in [84] in a SRAM based implementation. This gave to the proposed implementation a direct benchmark to have a first evaluation of skyrmion technology for LIM devices. The proposed implementation represents a first step in the direction of a complex LIM architecture based on skyrmion. The basic memory cell is indeed capable of doing basic elaboration on the stored information and ensure that the stored information is maintained inside the memory without data loss. The results showed in this chapter have been presented in the article [85].

## 5.1 Algorithm

The Minimum/Maximum search algorithm allows to identify in a set of words the minimum or the maximum value. The pseudo-code describing the algorithm is the following:

The algorithm exploits the independence of the filter operation to execute in parallel the operation and speedup the overall computation. In case the elaboration unit is able to process at the same time the complete set of words interested by the computation the latency can eventually be independent from the number of words elaborated. This last case recalls the logic-in-memory scenario in which the algorithms implemented can benefit from a high level of parallelism. The search of the maximum starts with the initialization of the the enable signals for all the words

```
  Data: The input of the algorithm is the bit matrix containing all the bits
        stored in memory.
  MAXIMUM():
1 for $i$ = 1 downto $N_{words}$ do
2     $\mathrm{enable}_{\mathrm{N_{bit}},\mathrm{i}}=1$
  end
3 for $j = N_{bit}$ downto 1 do
4     $x_\mathrm{j} = (bit_{\mathrm{j,1}} \cdot enable_{\mathrm{j,1}}) + (bit_{\mathrm{j,2}} \cdot enable_{\mathrm{j,2}}) + ... + (bit_{\mathrm{j,N_{word}}} \cdot enable_{\mathrm{j,N_{word}}})$
5     Parallel for $k$= 1 to $N_{words}$ do
6         $enable_{\mathrm{k,j}} = (enable_{\mathrm{k\text{-}1,j}} \cdot bit_{\mathrm{k,j}}) + (enable_{\mathrm{k\text{-}1,j}} \cdot (\overline{x_\mathrm{j}}))$
      end
  end
7 return $enable_{\ 0}$
```

**Algorithm 1:** Maximum algorithm

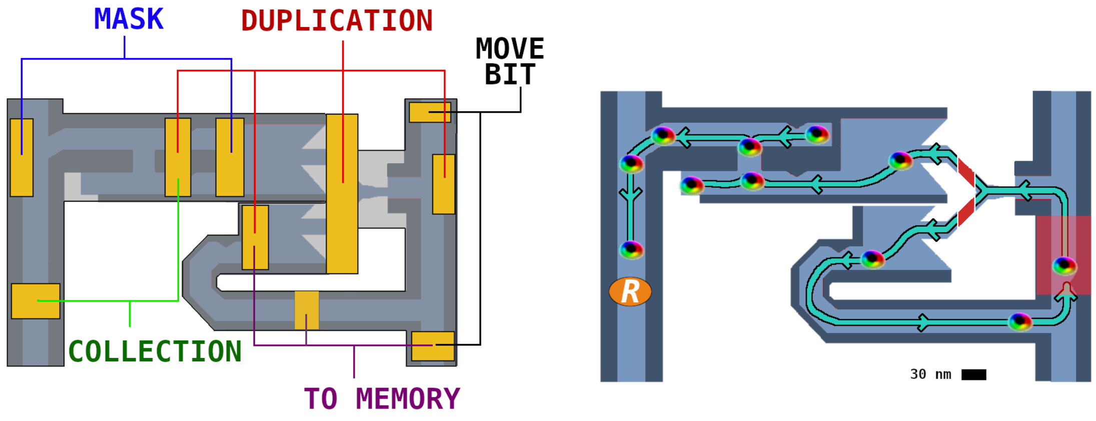
composing the set to 1. This set of signals will determine if a word is valid for the maximum (minimum) search and can continue in the evaluation in the next steps of the algorithm. After the initialization, the execution enters in the main loop. The first bit for the comparison is then selected starting from the Most Significant Bit (MSB). Now, at line 4, the validity of the comparison is evaluated. The variable $x$ indicates if the current stage of the algorithm should be considered or not. This condition is verified only in case all the bits in the current position $j$ are 0 (1). If the stage is valid, the enable signal for every word in the set is computed. The enable signal will determine if the current word is eligible to be the maximum (minimum) for the computation or if it should be discarded. The enable signal will be equal to 1 only in case the current bit is 1(0) and the enable signal for the current word is 1. The validity evaluation and enable signal calculation will be computed until all the bits have been examined. The last enable computation will then signal the position of the maximum (minimum) in the array.

In the next section the architecture implementing the algorithm will be presented, starting from the memory cell.

## 5.2 Memory Cell

The kind of memory used for the design presented in this chapter is the skyrmion racetrack memory. The basic track used for the presented implementation is the one used for the CAM cell implementation, section 3.2.1. The skyrmion in the chosen structure moves inside a curb of the ferromagnet. The information then is stored in the racetrack in defined portions of the track that will be considered, from here on, the basic memory cells of the LIM device. Compared to a common racetrack, the proposed design allows to perform logic operation on the skyrmion simply pushing it in a lateral track, placed on the left side in correspondence of the storing position of the bit. The proposed design is showed in figure 5.1b. The vertical track showed in the image is the portion of the racetrack storing the bit of information. The skyrmion moves along the track as long as memory operations are required. To enable the operation the memory is then connected on the left to a patterned structure designed to elaborate the information. The skyrmion will start its elaboration entering the top lateral track. Here the fork structure allows the duplication of information. The structure, proposed in [56], transforms the skyrmion in a domain wall pair and duplicates it in this form. The two copies of information are then translated back into skyrmion form to be further processed. After the duplication gate the information spitted in two follows two different paths. The track on the bottom allows the restore of the original information in the track to avoid data loss. The track on top executes the masking operation required by the algorithm. Finally after the skyrmion went through the top gate the result is collected in the vertical track on the left to be read and further elaborated. Many notch are placed along the track to ensure the correct synchronization of information. To read out the information a read-head is present in the left track. To reduce the collection latency with respect to a common racetrack memory the showed read-head is present every two cells. In addition a write-head is placed in correspondence of the mask operation to generate the enable signal in case needed. In figure 5.1b an example of a complete operation is presented. The figure shows the operation in case a 1 is stored

To support the operation a dedicated set of contacts has been designed. The presented contact distribution, shown in figure 5.1a has been designed to concentrate the current only in the section of the track interested by the computation. Especially in case of duplication the current densities involved in the transformation can be high as $50 \times 10^{10}\,\mathrm{A\,m^{-2}}$. Limiting the current generation to a limited portion of the track, as will be discussed in section 5.7, can impact in a positive way on the overall power and energy requirement of the circuit. The contacts activation happens in groups as shown in figure. When a particular operation is needed the contacts in the set are connected to supply and ground to produce the required current stimulus. The contacts are connected to the W layer. The current density required for the motion of the skyrmion is equal to $1 \times 10^{10}\,\mathrm{A\,m^{-2}}$. When the skyrmion stops at the notch along the circuit, a pulse of $0.3\,\mathrm{ns}$ and intensity $20 \times 10^{10}\,\mathrm{A\,m^{-2}}$ is used to depin it from the notches. Finally a current density of $40 \times 10^{10}\,\mathrm{A\,m^{-2}}$ is needed to process the skyrmion in the duplication gate.



(a)

(b)

Figure 5.1: In a), the contacts for current generation in the memory cell. The grey area indicates the tungsten layer. For every operation a different set of contacts is involved to generate the required local currents to move the skyrmion. In b), the path followed by a skyrmion entering in the processing zone in case a 1 is stored in the cell and the mask bit is set to 1. Image extracted from [85]

## 5.3 Cell operation

In this section, the operation of the cell will be presented and commented. The complete elaboration of the skyrmion is presented in figure 5.2

The operation has been divided in four steps:

1. The operation of the circuit starts from the memory track. Here the skyrmion is pushed with a vertical current in the direction of the input track for the elaboration zone. (Figure 5.2a);

2. The skyrmion is pushed with an horizontal current pulse inside the fork shaped nanotrack. The skyrmion is here converted in a domain wall pair. The current now pushes the newly formed domain along the fork causing the generation of two domains with the same polarity. Both the domains are then pushed to the edge of the structure. Here pushed by a second current pulse the domains are depinned from the constriction and converted back into two skyrmions. Both the skyrmions stops at two synchronization notches. In addition the auxiliary skyrmion if needed is nucleated in the top track to be ready for the next elaboration.(5.2b)

3. The operation is now different for the two information skyrmions. The information on top is depinned by means of a current pulse to be elaborated.

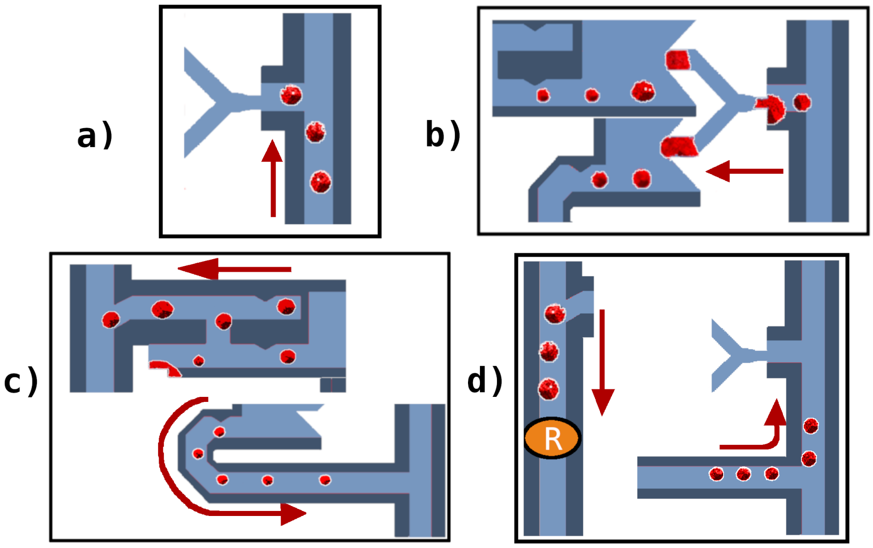


Figure 5.2: In figure the four phases of elaboration of the stored skyrmions. In a) the information is moved from the memory track into the duplication gate. In b) the information is duplicated by means of Skyrmion-DW conversion. In c) the information is processed on top and moved back to the memory cell on bottom. In d) the result of the operation is read on the left, the information is restored into the cell on the right. Image extracted from [85]

The gate used for elaboration is the AND/OR gate proposed by Chauwin et al. [55] presented in section 2.2.2. The result is collected on the top track and sent to the vertical track for the final collection. The presence or absence of a skyrmion on the top track of the gate is equivalent to the AND operation between the information encoded by the two skyrmions. The result of the bottom operation is annihilated against an unprotected edge. The copy of information on the bottom is depinned from the notch and routed to be reinserted in the memory track.(5.2c)

4. Finally the result of operation on top is directed to the read head where the result is collected to be further processed. The information on the bottom of the cell continues its path in the direction of the memory track. The information is at the end of the process correctly restored in the memory. (5.2d)

## 5.4 Memory organization

Some considerations on the memory organization will be presented in this section. The words for a correct function of the LIM cell are memorized along different tracks. Figure 5.3 shows how the read and write heads are organized along a single track. A single write head is placed at the beginning of the track. The write operation, like other racetrack memories is serial: the bit of information is nucleated and then shifted by one position; the operation is repeated until the complete word is written in memory. The bits of information occupies now precise position inside the memory array. The read operation is then executed by means of the read heads placed along the track. The reading operation is executed by means of a Magneto Tunnel Junction (MTJ) in which the free layer is the ferromagnet hosting the skyrmion memory. For other details, 2.2.2. Due to the serial nature of both write and read operations, the time required by an operation with a single write or read head is equal to $N_bit * T_cycle$. The presence of multiple read head allows to reduce the read latency of the memory. This require a more complex read architecture where multiple read circuit, 5.3 will write at the same time on the output shift register. After the read operation, the word should be restored in the initial position, ready to be read again or elaborated. The insertion of multiple read-heads allows also to diminish the latency of this restore operation.

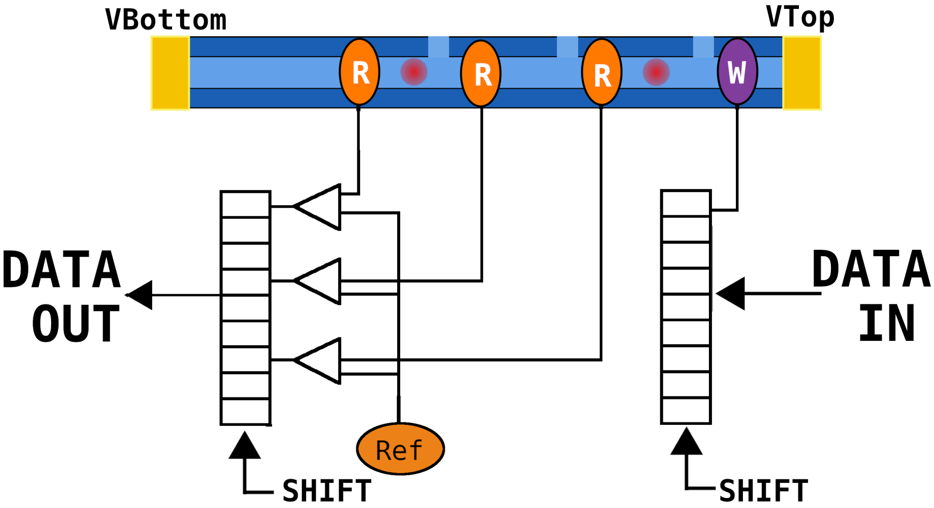


Figure 5.3: Read and write mechanism. The track has more than one read head to reduce the read operation latency. In order to sense the skyrmion presence under the read-head, the value produced is compared with a reference Image extracted from [85]

## 5.5 Logic-in-memory

In this section the complete logic-in-memory architecture will be presented. The complete structure of the LIM device is shown in figure 5.4.

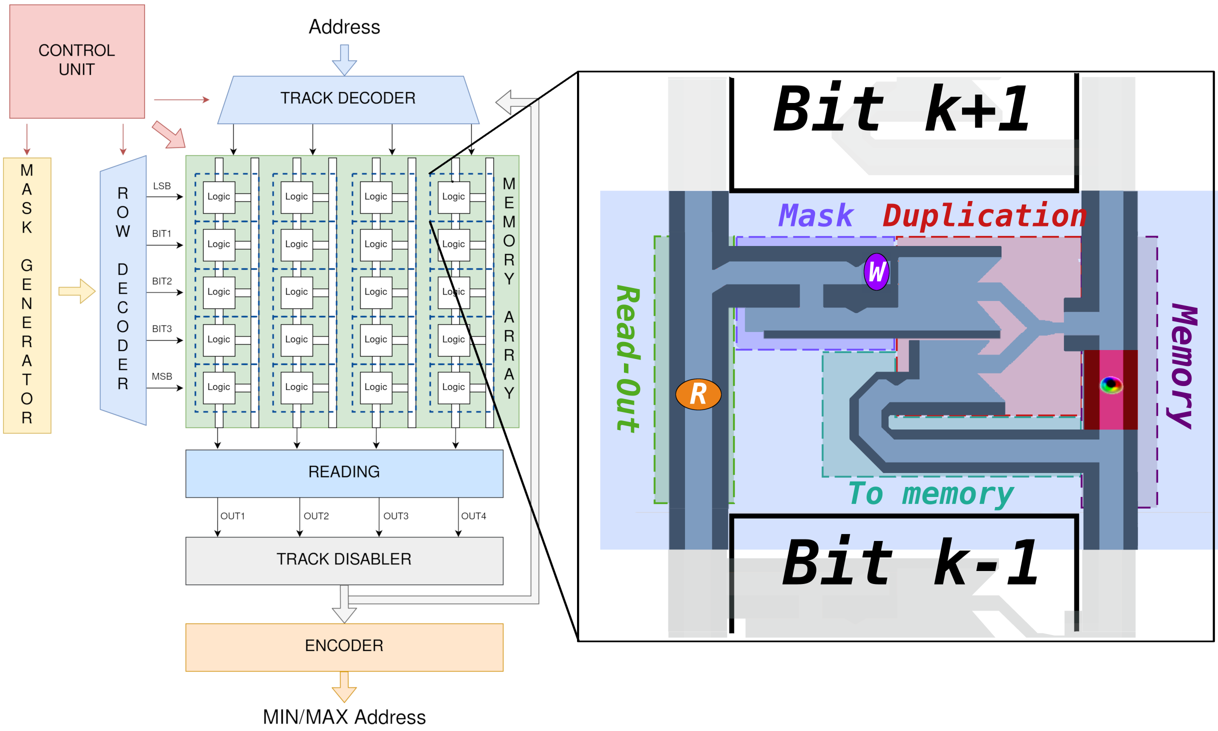


Figure 5.4: logic-in-memory architecture schematic. On the left the block diagram of the complete circuit. On the right the basic cell of the array. The skyrmion is stored inside the memory in correspondence of the red dot.Image extracted from [85]

The memory array communicates with a set of CMOS logic blocks to realize both the memory and the logic function. The surrounding logic allows both the interface with the external environment and the correct operation of the additional logic. The TRACK DECODER allows to control the enable signal of the different tracks. The TRACK DECODER during the memory operations reads the external address enabling only the memory track required. During the logic-in-memory operation the control is received by the TRACK DISABLER, removing from the computation all the tracks not anymore interested by the maximum search operation. The other direct control of the array is the ROW DECODER. This block in synergy with the track decoder allows to select the precise row interested by the computation. The row decoder receives its commands from the MASK GENERATION block. The MASK generation block is the one in charge of generating the correspondent signals for the ROW DECODER to correctly activate the required rows in the computation. The MASK GENERATION in the implemented algorithm will act as a shift register activating sequentially the rows starting from the MSB position down to the Least Significant Bit (LSB). The READING block at the periphery of the array groups all the required circuitry to read out the values present at the different read heads distributed in the memory and provide the output port of the memory and to the other control circuit of the array. The value read by the READING block are then sent to the TRACK DISABLER that is in charge of determine which track will continue in the computation for the TRACK DECODER. When the algorithm is over the ENCODER will read the TRACK DISABLER signal and the address of the minimum or the maximum will be produced at the output of the circuit. If multiple enable signal are equal to 1 when the algorithm is over the encoder will apply a priority rule signaling the maximum with the lower address. All the operations are directed by a single control unit that keeps track of the steps of the algorithm and coordinates the operation of all the logic block to realize both the memory and the LIM functionality.

### 5.5.1 Maximum/Minimum Search Operation

When the search operation is required the initial bit is loaded in the MASK GENERATION block. The starting MASK, sent to the ROW decoder, will select the cells correspondent to the MSB position of the array for the first elaboration. The FSM now exits from the idle state entering in the main loop, as shown in figure 5.5. The FSM executes in the main loop 5 cycle to elaborate the information. The first four cycle are the ones producing the control for the basic cell as shown in section 5.3. During this operation the control unit allows the correct generation of the stimuli in the cell. After the elaboration is completed in the fifth cycle the value read by the reading block is sent to the TRACK DISABLE block that generates the new set of enable signal for the tracks that are still valid for the computation. Now the MASK is shifted by one position and the cycle can start again repeating the five cycles for a number of times equal to the length of a single word. Finally when the final count is reached, the MAX/MIN state is reached. In this state the ENCODER is triggered and the maximum position is generated at the output. The operation of the FSM is over and the machine goes back to the idle state.

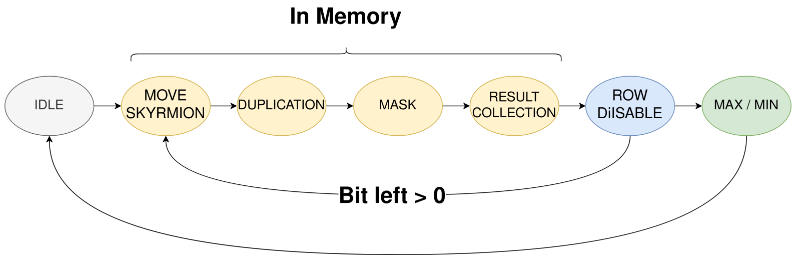
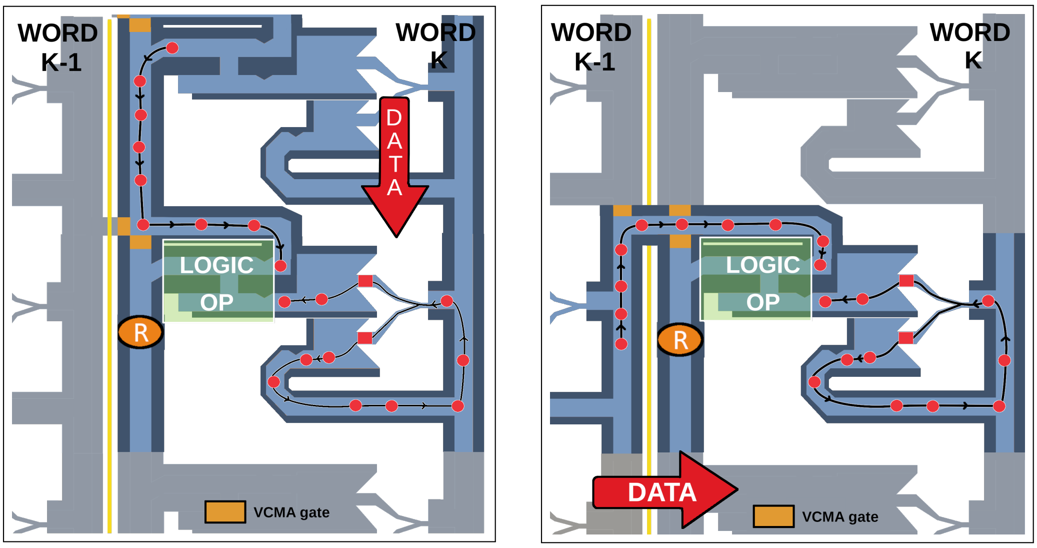


Figure 5.5: FSM State diagram for Maximum/Minimum search Image extracted from [85]

The presented control unit controls also the reading and the writing operation.

### 5.5.2 Additional logic operations

The proposed logic-in-memory approach can be adapted to many other algorithms. In particular exploiting the confined nature of the skyrmion the cells can be interconnected with additional tracks with respect to the proposed basic implementation. A possible modified version is shown in figure 5.6



(a)

(b)

Figure 5.6: Logical operation between rows and columns. a) Column-wise operation b) Row-wise operation. [85]

The defined interconnections increase the degrees of freedom in the computation. The example showed in figure 5.6a enable the execution of column wise operations. The result of the operation of a single cell can be forwarded to the next cell to obtain a chain operations with the data. Another possible interconnection is the one showed in figure 5.6b. With this topology the skyrmions coming from neighbor columns can be elaborated together in the same cell to produce additional logic. In the next chapter, exploring a cryptography algorithm that requires more complex communications between cells these kind of interconnection will be heavily exploited to realize the complete algorithm in memory. Finally to further expand the actual implementation different rows or columns can be specialized implementing different operations. As shown in [55], many kind of logic can be realized with the presented gates. The distribution of logic inside the memory can help also in maintaining the density of the memory high even including the logic inside the memory cell. In this scenario indeed a logic operation can be the result of the information traveling through different portion of memory.

## 5.6 Methods

To verify the proposed structure the device was simulated in an heterogeneous environment to verify both the functionality and the performance of the complete LIM circuit. The cell was simulated using the Mumax$^3$ [19, 53]. The single cell was meshed with a basic cell with dimensions $1\,\mathrm{nm}^3 \times 1\,\mathrm{nm}^3 \times 0.5\,\mathrm{nm}^3$. The parameters used for the simulation are reported in Table 5.1. The micromagnetic simulation was performed to verify the correctness of the design and to evaluate the timing performance of the circuit.

Table 5.1: Parameters used for micromagnetic simulations and current distribution computation

| SIMULATION PARAMETERS | | |
|---|---|---|
| Saturation Magnetization [52] | 1e6 | $\mathrm{A\,m^{-1}}$ |
| Uniaxial Anisotropy Constant [52] | 8e5 | $\mathrm{J\,m^{-2}}$ |
| Exchange Stiffness [52] | 2e-11 | |
| Damping constant [75] | 0.03 | |
| Spin Hall Angle [63] | 0.4 | |
| Film resistivity [63] | 165 | $\mu\Omega\,\mathrm{cm}$ |

To verify the contact design and the power required by the circuit to produce the movement of current, the stack was then modeled and simulated with ElmerFEM software [86] with the static current module. This tool allowed a precise study of the aforementioned quantities to better understand the impact on the overall performance of the cell operation. This last method was chosen for the non-trivial configuration of the conducting path when the current is produced. As will be shown in the next section the parasitic currents were also evaluated. The resistivity and the spin hall angle parameters employed for the micromagnetic and static current study were derived from [87].

The additional logic of the circuit, described in VHDL was tested with Mentor Questasim to verify its correctness and then synthesized with Synopsys Design Compiler on a 15nm Nangate technological library to evaluate its performance. To obtain a better estimation of the power consumption, the extracted power was estimated using a back-annotation approach. A sample application was then set up and the power consumption related to that case was extracted. The sample application is a complete execution of the algorithm in which all the rounds are valid and only a single word is removed at every cycle.

## 5.7 Results

In the following the performance and the simulation results of the device will be shown and compared with the implementation proposed by [84]. The implementation in [84] is a complete CMOS implementation of the algorithm based on a modified SRAM memory that takes advantage of some additional transistors on the single cell to execute the filtering and collection of information. The similarity in the implementation and the same approach applied to the presented architecture allow to better evaluate the strengths and weaknesses of the proposed implementation. The performance of the LIM circuit are shown in table 5.2, 5.3 and 5.4

Table 5.2: AREA and TIMING, WORD WIDTH = 64 bit

| Words | Area (µm$^2$) | | | Frequency (MHz) | | Latency (ns) | | |
|---|---|---|---|---|---|---|---|---|
| | Proposed | [84] | Ratio | Proposed | [84] | Proposed | [84] | Ratio |
| 2048 | 42307 | 89665 | 0,47 | 285* | 1785 | 1120 | 108 | 10,30 |
| 4096 | 95036 | 181694 | 0,52 | 285* | 1041 | 1120 | 183 | 6,20 |
| 8192 | 190056 | 368489 | 0,51 | 285* | 571 | 1120 | 373 | 3,00 |
| 16384 | 381625 | 747729 | 0,51 | 285* | 298 | 1120 | 647 | 1,74 |
| 32768 | 748076 | 1516103 | 0,49 | 285* | 149 | 1120 | 1293 | 0,86 |
| 65536 | 1427936 | 3045940 | 0,46 | 285* | 75 | 1120 | 2560 | 0,44 |
| * limited by maximum array frequency | | | | | | | | |

The proposed architecture can reach a maximum frequency of 285 MHz. The dimension of the array was varied up to a value of 64 Kwords. The critical path in all the tested combinations was found in the skyrmion logic. In particular the longest path the skyrmion has to travel sets the speed of the complete architecture. The longest paths in the designed cell are represented by the return path the copy of the original skyrmion travels to return back in the memory and the distance between consecutive cells in the memory. For the additional logic the slowest path is determined by the encoder that has to produce the final result for the computation. The fact that the critical path of the logic is lower than the critical path of the skyrmion array allows to relax the frequency constraint of the logic reducing the overall power consumption. From the comparison on the maximum achievable frequency it is possible to notice that the proposed architecture is able to reach higher values after the size of memory increase over 16 Kword. Nevertheless it is important to underline that the proposed architecture has still an higher latency up to 32 Kword because the proposed algorithm requires two additional cycles per every iteration of the algorithm with respect to the CMOS implementation. The area occupied by the LIM device has been computed summing up the area occupied by the skyrmion array and the control logic. The former was extracted from the physical structure shown in figure 5.1b, the latter was extracted from the synthesis of the written code. As shown in table, the area occupied by the LIM array based on skyrmion technology is half of the one occupied by the implementation of the CMOS. This is due to the fact that the memory basic memory element in the racetrack memory is represented by the single track that occupies while the CMOS implementation is based on a SRAM cell that requires at least 6 transistor for the realization of the basic cell. After the evaluation to better appreciate the comparison between the two architectures the Delay-Area product has been evaluated. The results are shown in figure 5.7. This quantity is a measure of the cost of the implementation of the circuit. From the data it's possible to see how the choice between the two technology, driven by the cost of the implementation, is not obvious and depends strongly on the size of the required memory. In case of small implementations indeed the tremendous speed of CMOS prevail but with the increase of the array size the speed becomes comparable and even worse leading to higher size values in which the ratio between the two products is completely reversed in favor of the skyrmion implementation.

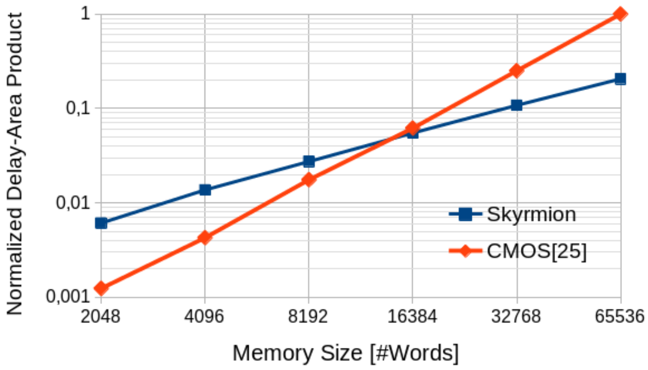


Figure 5.7: Delay area product with respect to the memory size of the proposed skyrmion implementation and CMOS [84]. Image extracted from [85]

Table 5.3: Power consumption of a single memory cell for each phase

| Phase | Power [uW] |
|---|---|
| Move Skyrmion | 0.07 |
| Duplication | 21.67 |
| AND | 6.44 |
| Collection | 0.14 |

After the evaluation of the time and area parameters the energetic performance are shown. The evaluation of these parameters as explained in the 5.6 section has been evaluated with static conduction simulations for the cell and by means of the back-annotation process for the logic part. Regarding the power required by the cell to operate the evaluation has been done for every operation of the cycle. As

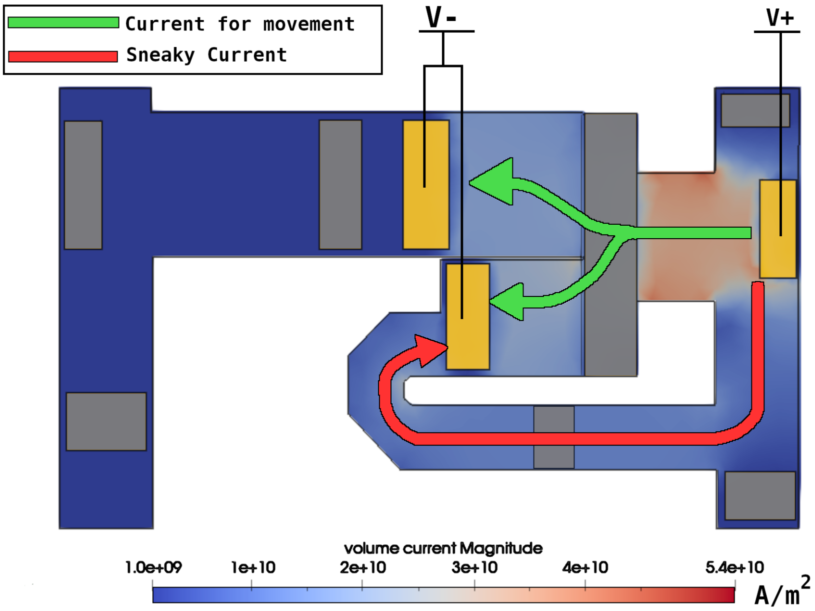


Figure 5.8: Current density produced during duplication operation. The current path highlighted in green is the one required to move the skyrmion. The current path in red indicates the additional path for the current not needed for the current. This additional paths increase the power consumption in the operation. Image extracted from [85]

shown the most demanding operations are the duplication and the AND operation. The high power consumption of the duplication phase is justified by the fact that to correctly operate the duplication phase needs two pulses, one to depin and convert the skyrmion into a domain wall pair and the second to depin both the domain wall pairs from the constriction and translate them back into skyrmions. In addition, as shown in figure 5.8 both the operation have a non-negligible amount of current that is generated by the additional conductive paths and by the different potentials required across the circuit to produce multiple skyrmion movements at the same time. This parasitics are mainly due to the fact that the skyrmion paths are conductive. For this reason every circuit implementing feedback loops, as the one proposed will have additional current paths increasing the power consumption of the circuit. In the case of duplication the V+ and V- are not only connected through the intended skyrmion path, the fork structure but also through the feedback loop. In table 5.4 the value of mean power consumption and energy per bit are reported.

The reported values refers to 64 bit words. With regard to the power consumption the CMOS implementation showed to perform always better than the proposed skyrmion LIM implementation. This is mostly due to the very high power consumption every cell needs to operate. The energy consumption in the same way shows

Table 5.4: POWER AND ENERGY - WORD WIDTH=64 bit

| Words | Power (mW) | | Energy per bit (pJ/bit) | | |
|---|---|---|---|---|---|
| | Proposed | [84] | Proposed | [84] | Ratio |
| 2048 | 34 | 25 | 327 | 20 | 16,35 |
| 4096 | 66 | 36 | 320 | 25 | 12,80 |
| 8192 | 74 | 54 | 193 | 38 | 5,00 |
| 16384 | 133 | 85 | 177 | 52 | 1,92 |
| 32768 | 227 | 148 | 157 | 90 | 1,74 |
| 65536 | 574 | 270 | 153 | 164 | 0,93 |

how the overall efficiency is higher for the CMOS circuit. Nevertheless it is possible to see how the trend in energy efficiency is negative for the skyrmion implementation while is positive for the CMOS implementation. This suggests that for even bigger arrays the skyrmion implementation is expected to perform sensibly better. The reason behind the different trends can be found in the way, the working frequency of the circuit is scaling differently for CMOS and skyrmions. While the CMOS implementation lowers the frequency of operation and consequently increase the overall latency of the algorithm, the skyrmion implementation maintains both the parameters stable allowing the improvement of the energy efficiency. The negative trend of the energy efficiency is expected not to continue indefinitely but to stabilize and then start to increase in a similar way of CMOS. This because the critical path of the memory array will be at some point surpassed by the critical path of the surrounding CMOS logic.

## 5.8 Final considerations

The implementation of a LIM architecture for minimum/maximum search based on skyrmion technology proved to be valid even compared with a CMOS equivalent implementation. In particular the architecture showed good performance for high levels of parallelism and a very small occupied area compared to the same implementation realized in CMOS. At the same time for small parallelisms the reference CMOS architectures shows better performance thanks to low delay the architecture can achieve. Skyrmion technology on the other side is limited up to a point in which the reduced speed of skyrmions is comparable with the other architectures. In addition, the architecture presented addresses also some important issues of LIM architectures based on skyrmions, as the duplication of information and the current generation inside the circuit that proved to be critical for the overall performance of the circuit. These two aspect proved to be the most problematic in reaching low power consumption for the architecture.

# Chapter 6

# Advanced Encryption Standard

After discussing the minimum-maximum algorithm, the next skyrmion LIM implementation explored is the Advanced Encryption Standard (AES) cipher. The proposed implementation can encrypt plain text stored in memory without the need to read it out. The information is only locally read and translated in an electrical signal during the SubBytes operation. The goal behind implementing such a complex algorithm in memory is twofold: on one hand there is the interest in verifying the achievable performance of a complex logic architecture. On the other hand, the goal of the implementation is to explore the design of a complex algorithm that requires at the same time interaction between stored information and elaboration on the stored data. This implementation will allow to better appreciate the strength of the technology for logic implementation and at the same time understand what are the constraints in the design of complex architecture.

AES is the current standard for cryptography. Elaborated in 1998 by Joan Daemen and Vincent Rijmen [88] to substitute the Triple Data Encryption Algorithm (TDES) cryptographic algorithm, the Advanced Encryption Standard is used in many applications to keep the information secure, especially in the Internet-of-Things (IOT) environment. The AES uses a series of logic operations on the input data to obtain an encrypted version of the information. These logic operations refer to specific transformations in the Galois Field, a specific finite-elements field applied extensively in cryptography. The same operations inversed in the Galois Field can then restore the plain version of the original information. Some of the logic operations depend on the second input of the cryptography algorithm, a key used to produce a unique transformation of the original data. The algorithm uses the key to influence the cipher after a series of logical operations called Key Expansion. These operations link the encrypted data with the specific input key. The length of the key determines then the length of the algorithm and, consequently, the cipher's strength.

## 6.1 AES algorithm

The input of the AES algorithm is called **state**. The state is composed from 128-bit blocks divided into 16 words of 8 bits. The state is usually represented as a 4x4 matrix of 8-bit words as shown in figure 6.1. [89] This representation has a specific meaning in the execution of the algorithm. The basic algorithm elaborates the state often in matrix form following basic rules of matrix algebra.

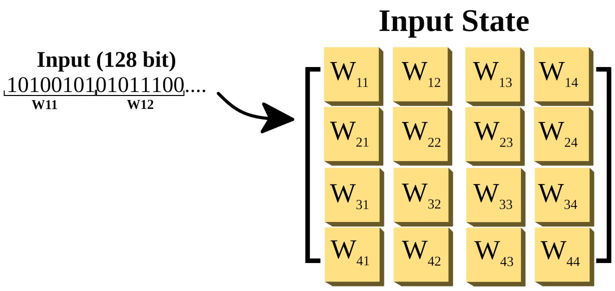


Figure 6.1: AES input state matrix representation

The AES algorithm is performed through 4 main operations:

- Sub Bytes
- Shift Row
- Mix Columns
- Add Round Key

After a first Add Round Key operation, the four operations are executed in the order shown in figure 6.2. Every repetition of the four operations is called round. In order to encrypt the plain text, the state is elaborated in many rounds to complete the encryption. In AES algorithm, all the rounds are equal except the last in which the Mix Columns operation is not repeated. Depending on the length of the key, the cipher algorithm can have a different number of rounds. The three standard key lengths are 128-bit, 196-bit, 256-bit. To these three lengths, correspond respectively a length of 10, 12 and 14 rounds. The length of the key also define the strength of the encryption, where a longer key determines a stronger cipher. [89]

### 6.1.1 Key expansion

Before a round can be executed, the key should be expanded to produce the correct inputs for the algorithm execution. The goal of the key expansion is to produce from a key of length 128,196 or 256 bit respectively 11,13 and 15 matrices

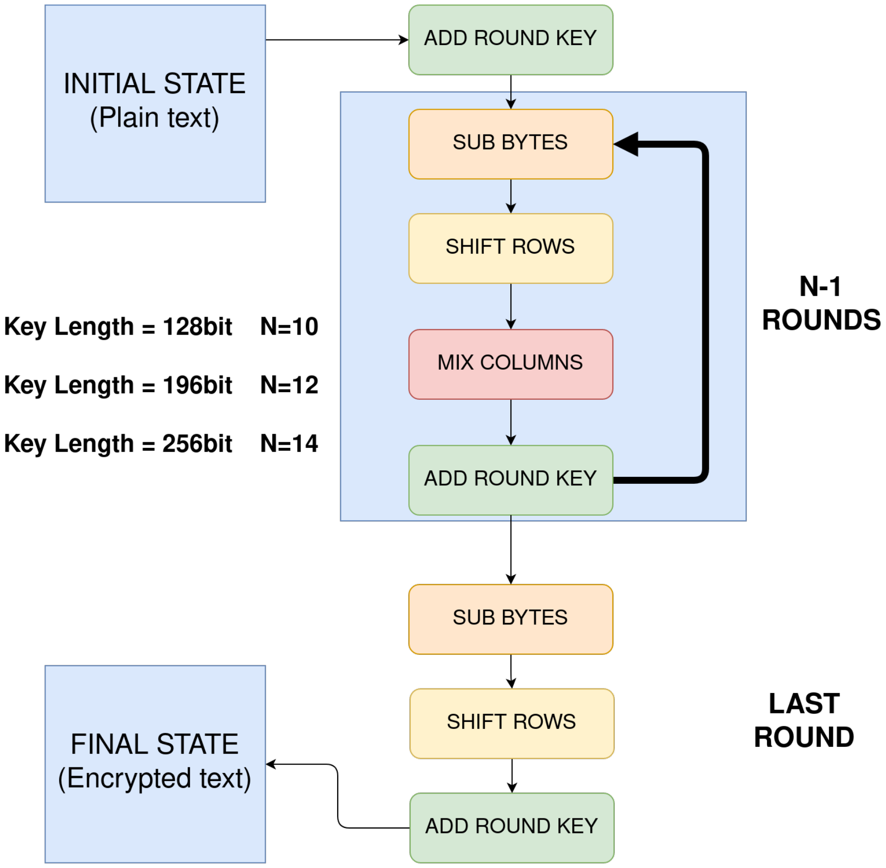


Figure 6.2: AES encryption algorithm general block diagram

to be used in the AddRoundKey operation. The algorithm for the key expansion is showed

where $RotLeft()$ indicates a rotation of one bit on the left and $SubWord$ indicates a substitution of the word using the S-Box. Finally RCon is a constant vector.

The input key is used directly to produce the first 4x4 round key. After this first passage, the rows of the following round keys are produced in two passages.

1. The last computed word, correspondent to the last element of the previously computed row is elaborated ($K15$ in the example). The algorithm of the variation element $g$ follows a path similar to the one applied to the input state. At every round: the word is rotated, the result is substituted using the S-Box of the SubBytes stage and the result is xored with a constant matrix.

2. After this elaboration, a chain of xor operations is executed to produce a new row for the next round key. The chain of xor starts summing the first element of the row with the variation element $g$. Then a XOR operation between this

```
KeyExpansion (byte key[16], word w[44]):
word temp;
for i = 0; i < 4; i++ do
    w[i]= (key[4i], key[(4i)+1], key[(4i)+2], key[(4*i+3)])
end
for i = 0; i < 4; i++ do
    if i mod 4 = 0 then
        // First word of the row
        temp = SubWord(RotLeft(temp)) XOR Rcon[i/4];
    else
        // All the other rows
        temp = w[i-1];
    end
    w[i] = w[i-4] xor temp
end
```

**Algorithm 2:** Key Expansion algorithm for a 128-bit key

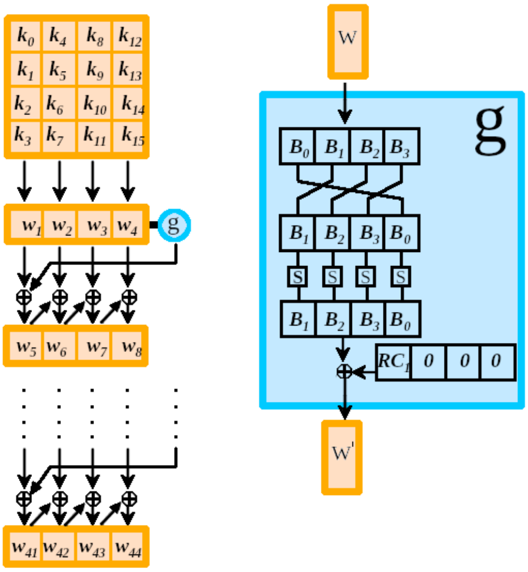

$k_0$ $k_4$ $k_8$ $k_{12}$
$k_1$ $k_5$ $k_9$ $k_{13}$
$k_2$ $k_6$ $k_{10}$ $k_{14}$
$k_3$ $k_7$ $k_{11}$ $k_{15}$
$w_1$ $w_2$ $w_3$ $w_4$
g
$w_5$ $w_6$ $w_7$ $w_8$
$w_{41}$ $w_{42}$ $w_{43}$ $w_{44}$
W
g
$B_0$ $B_1$ $B_2$ $B_3$
$B_1$ $B_2$ $B_3$ $B_0$
S S S S
$B_1$ $B_2$ $B_3$ $B_0$
$RC_1$ 0 0 0
W'

Figure 6.3: Key expansion algorithm procedure. The example shown is referred to a 128-bit key producing 44 words from the cipher key

last element and the corresponding in the previous row is executed. The chain continues until the row is complete.

The row generation explained above is repeated a different number of times depending on the length of key to produce a sufficient number of words to be fed to the add round key operations executed during the algorithm. For 128 bit the words required are 44, for 196 bits the words are 64, for 256 bits the words are 64.

### 6.1.2 Add Round Key

The first operation at the beginning of the encryption and the last of every round, is the Add Round Key. The round key of the current round, computed as explained in the previous section, is xored element by element with the current state to obtain a new state to proceed with the cryptography process as shown in figure 6.4. This XOR operation corresponds in the Galois field $GF(2^8)$ to the sum of two polynomials of grade 8 [89]. As it will be clear afterward, this is the only operation with the key expansion that changes, varying the algorithm's key input.

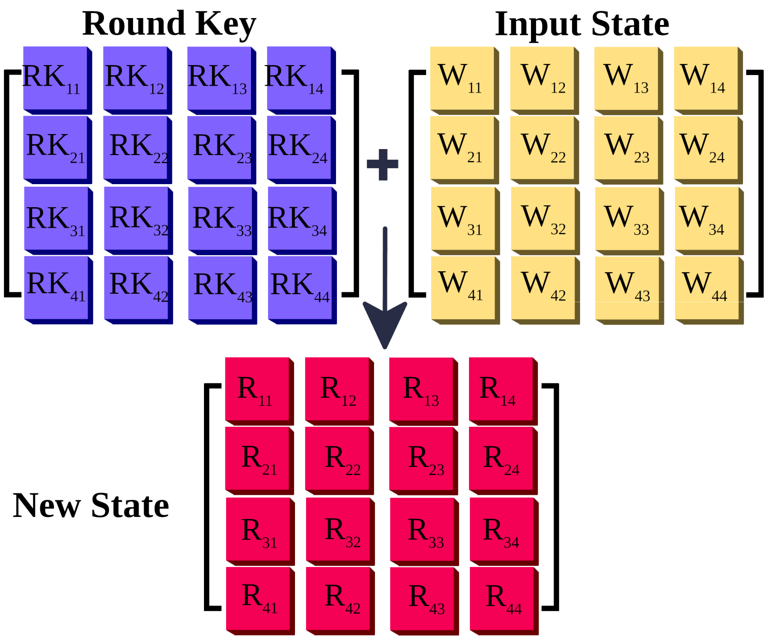
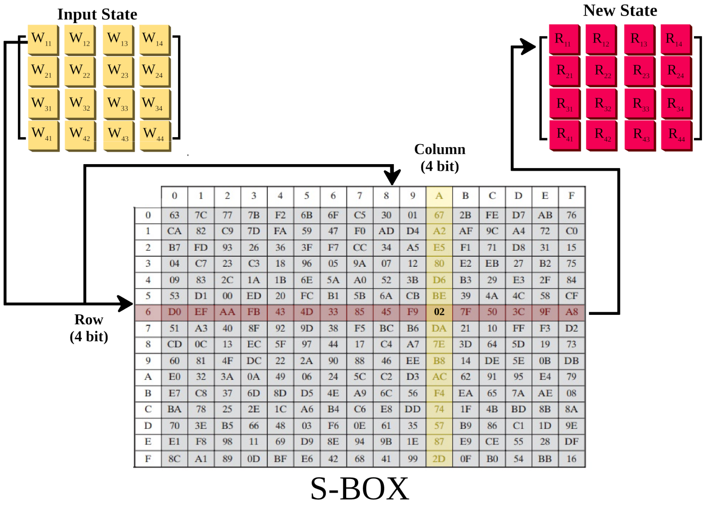

Round Key
$RK_{11}$ $RK_{12}$ $RK_{13}$ $RK_{14}$
$RK_{21}$ $RK_{22}$ $RK_{23}$ $RK_{24}$
$RK_{31}$ $RK_{32}$ $RK_{33}$ $RK_{34}$
$RK_{41}$ $RK_{42}$ $RK_{43}$ $RK_{44}$
+
Input State
$W_{11}$ $W_{12}$ $W_{13}$ $W_{14}$
$W_{21}$ $W_{22}$ $W_{23}$ $W_{24}$
$W_{31}$ $W_{32}$ $W_{33}$ $W_{34}$
$W_{41}$ $W_{42}$ $W_{43}$ $W_{44}$
New State
$R_{11}$ $R_{12}$ $R_{13}$ $R_{14}$
$R_{21}$ $R_{22}$ $R_{23}$ $R_{24}$
$R_{31}$ $R_{32}$ $R_{33}$ $R_{34}$
$R_{41}$ $R_{42}$ $R_{43}$ $R_{44}$

Figure 6.4: Add round key operation. RK matrix refers to the round key computed by the key expansion operation.

### 6.1.3 Sub Bytes

Sub bytes or Substitute byte transformation, is a table lookup that maps the current state to another following a fixed scheme. This operation is the first of each round. In the sub bytes operation, every word composing the current state is substituted by another value extracted from a fixed 16x16 words table called S-Box. To execute the substitution, the new element that will be placed into the row is chosen using the word as an index in the table. In particular, the four most significant bits are used as an index to identify the row. The four remaining bits are used to identify the column of the element that will substitute the current value. An example of the operation and the S-Box are shown in figure 6.5


Input State
$W_{11}$ $W_{12}$ $W_{13}$ $W_{14}$
$W_{21}$ $W_{22}$ $W_{23}$ $W_{24}$
$W_{31}$ $W_{32}$ $W_{33}$ $W_{34}$
$W_{41}$ $W_{42}$ $W_{43}$ $W_{44}$
New State
$R_{11}$ $R_{12}$ $R_{13}$ $R_{14}$
$R_{21}$ $R_{22}$ $R_{23}$ $R_{24}$
$R_{31}$ $R_{32}$ $R_{33}$ $R_{34}$
$R_{41}$ $R_{42}$ $R_{43}$ $R_{44}$
Column
(4 bit)
Row
(4 bit)
S-BOX

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63 | 7C | 77 | 7B | F2 | 6B | 6F | C5 | 30 | 01 | 67 | 2B | FE | D7 | AB | 76 |
| 1 | CA | 82 | C9 | 7D | FA | 59 | 47 | F0 | AD | D4 | A2 | AF | 9C | A4 | 72 | C0 |
| 2 | B7 | FD | 93 | 26 | 36 | 3F | F7 | CC | 34 | A5 | E5 | F1 | 71 | D8 | 31 | 15 |
| 3 | 04 | C7 | 23 | C3 | 18 | 96 | 05 | 9A | 07 | 12 | 80 | E2 | EB | 27 | B2 | 75 |
| 4 | 09 | 83 | 2C | 1A | 1B | 6E | 5A | A0 | 52 | 3B | D6 | B3 | 29 | E3 | 2F | 84 |
| 5 | 53 | D1 | 00 | ED | 20 | FC | B1 | 5B | 6A | CB | BE | 39 | 4A | 4C | 58 | CF |
| 6 | D0 | EF | AA | FB | 43 | 4D | 33 | 85 | 45 | F9 | **02** | 7F | 50 | 3C | 9F | A8 |
| 7 | 51 | A3 | 40 | 8F | 92 | 9D | 38 | F5 | BC | B6 | DA | 21 | 10 | FF | F3 | D2 |
| 8 | CD | 0C | 13 | EC | 5F | 97 | 44 | 17 | C4 | A7 | 7E | 3D | 64 | 5D | 19 | 73 |
| 9 | 60 | 81 | 4F | DC | 22 | 2A | 90 | 88 | 46 | EE | B8 | 14 | DE | 5E | 0B | DB |
| A | E0 | 32 | 3A | 0A | 49 | 06 | 24 | 5C | C2 | D3 | AC | 62 | 91 | 95 | E4 | 79 |
| B | E7 | C8 | 37 | 6D | 8D | D5 | 4E | A9 | 6C | 56 | F4 | EA | 65 | 7A | AE | 08 |
| C | BA | 78 | 25 | 2E | 1C | A6 | B4 | C6 | E8 | DD | 74 | 1F | 4B | BD | 8B | 8A |
| D | 70 | 3E | B5 | 66 | 48 | 03 | F6 | 0E | 61 | 35 | 57 | B9 | 86 | C1 | 1D | 9E |
| E | E1 | F8 | 98 | 11 | 69 | D9 | 8E | 94 | 9B | 1E | 87 | E9 | CE | 55 | 28 | DF |
| F | 8C | A1 | 89 | 0D | BF | E6 | 42 | 68 | 41 | 99 | 2D | 0F | B0 | 54 | BB | 16 |

Figure 6.5: Sub Bytes operation. The input word is substituted with another word using half of the word as row index and half of the word as column index

### 6.1.4 Shift Row

The second operation in every round is the Shift Row operation. In this operation, the state is considered again as a 4x4 matrix. This operation corresponds to a rotation of the row's elements constituting the state by a fixed amount depending on the row index. The rotation direction is left. The rotation steps are determined by the index of the row, assigning to the row on top index 0 and increasing it by 1, continuing with the other rows. Therefore the first row is not rotated, the second is rotated by one position, the third by two, and the last by three positions.

### 6.1.5 Mix Columns

The Mix Columns operation is one of the most complex operations of the algorithm and, as it will be shown later, one of the most energy and time consuming of the whole algorithm. The operation, differently from the others, is not repeated in the last round of the algorithm. In this operation, the state is multiplied with a constant matrix to obtain an updated state. The new state results from modulo matrix multiplication in Galois Field between the current state and the matrix shown in figure 6.6. Therefore, every element in the result's column is a function of all the elements of the previous state. In addition, if the shift row operation is considered, it is trivial to demonstrate that every word of the final state is a function of all the other words of the matrix. The multiplication and addition operations are executed in $GF(2^8)$.The constant matrix is shown in figure 6.6. The constant matrix was designed by the Rjndael authors to ensure a good mixing of the element of a column. An example of operation is shown in figure 6.6.

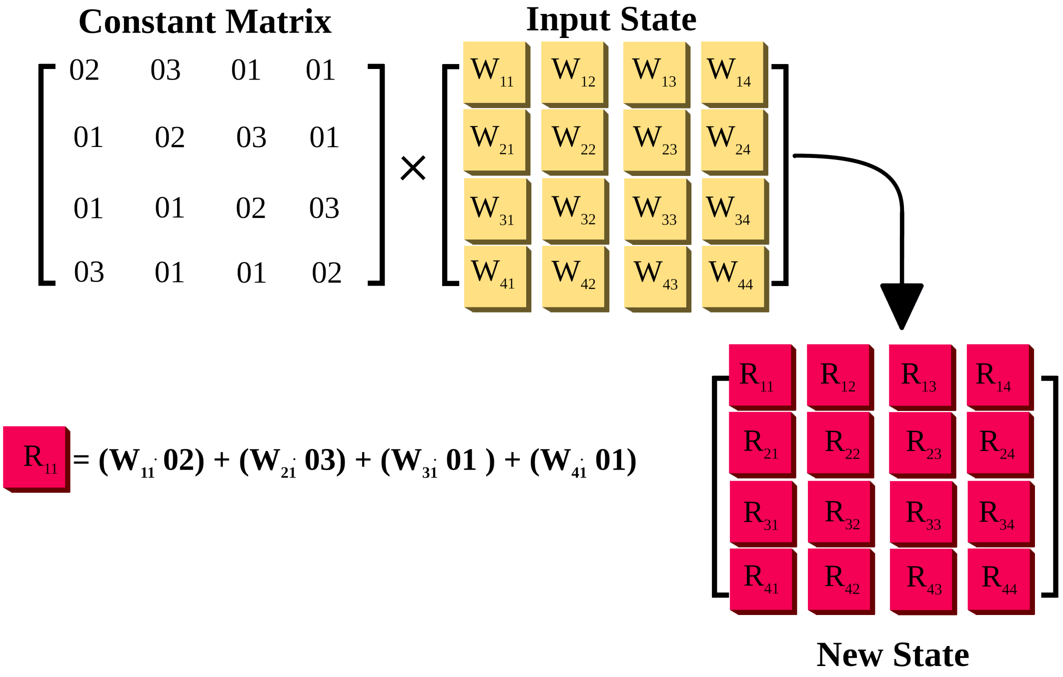


Figure 6.6: Mix column operation

A final note should be added to this particular operation regarding the binary operation required to execute the correspondent Galois field operations. The equivalent operation of the sum in the Galois field is the bitwise XOR operation. In order to execute multiplication by 02 and 03, which corresponds in the Galois field to multiplication by x and by x + 1 polynomials, shift and bitwise XOR operations are needed. In particular, to execute a multiplication by 02, the input of the operation can be shifted left by one position as a standard decimal operation, but, in addition, it should also be re-normalized. The shifted value is then additionally XORed with the constant hexadecimal value $1B$ (binary 0001 1011) if the MSB of the original word was 1.

The multiplication by three can be realized by taking advantage of the corresponding polynomial characteristics, $x + 1$. The operation can be executed as multiplication by two plus the sum to the original value as shown in the following:

$$(56) \cdot (03) = (56) + ((56) \cdot (2)) \tag{6.1}$$

This characteristic of the stage makes the operation required for encryption easier. This operation followed by an Add Round Key concludes a single round. As explained above, this schema is repeated at least ten times to obtain an encrypted text. Decryption operation follows a similar schema. The difference in the decryption is the order of the operations and in the coefficients used for the operations needed to obtain an inverse transformation of the state. In the following implementation, only the encryption procedure will be mapped to memory as the algorithm constitute an important case study for the purpose of the thesis. The proposed design was not designed to execute both the encryption and the decryption of the stored text.

## 6.2 AES logic-in-memory with skyrmions

### 6.2.1 LIM device structure

The structure of the complete LIM device is shown in figure 6.7.

The basic structure is the one of a traditional memory where the stored information can be read and written. In addition to the decoder and write and read controls, additional circuitry is needed to realize the logic functionality required by AES. In particular, when a new encryption process is triggered, the KEY EXPANSION block generates the round keys needed for the encryption, starting from an external key input. These values are then sent to the write control to be used in the LIM device during the different AddRoundKey operations. The circuit also has an external S-BOX circuit. This circuit is a lookup table that provides the required substitution of information for the SubBytes process. All the other operations required by the encryption are executed in the LIM array. A control unit directs all the operations. Depending on the required functionality, i.e., read, write or encrypt, it generates the required control signals to coordinate the operation of the different external units and the LIM device. A VCMA control is also present. The control of the VCMA gates in the array is fundamental for the correct routing and synchronization in the array.

Finally, additional skyrmion memories can be put directly in contact with the LIM array to increase its storage capabilities and give an alternative way to input information. This additional possibility for storage will be explained more in-depth in section 6.2.7.

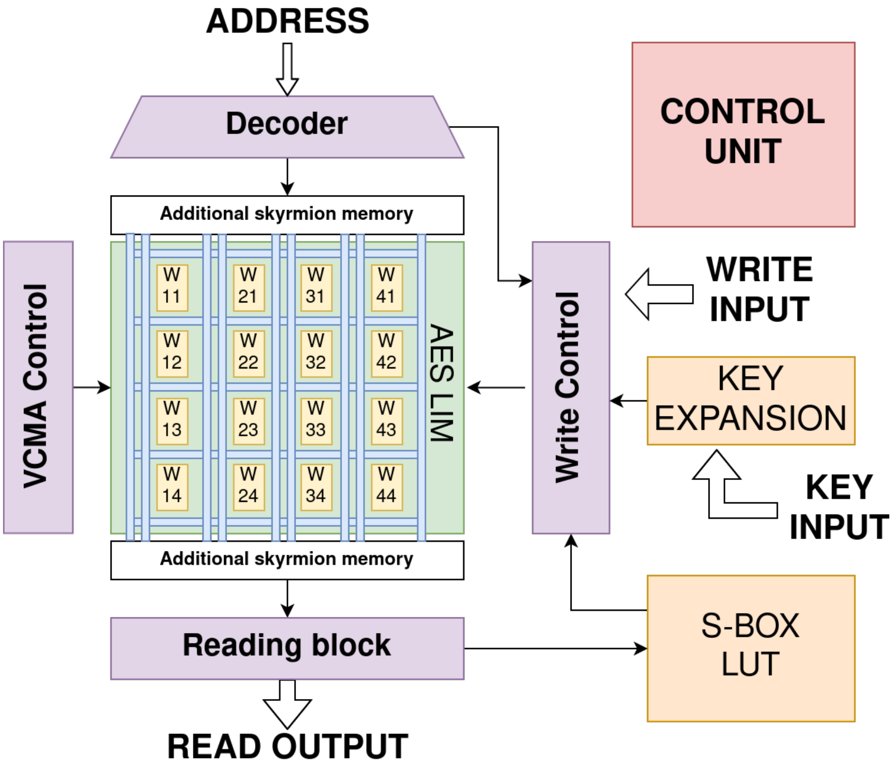


Figure 6.7: Block diagram of the complete LIM device.

### 6.2.2 Array Design

In translating the AES algorithm into a skyrmion logic-in-memory device, the starting point is the memory array that will host the LIM implementation. The first constraint for the LIM design is that the circuit should have the possibility to be integrated with skyrmion memories. Racetrack memories, in particular, are the reference for memories based on skyrmions. The racetrack memory is a nanowire in which skyrmions, encoding information, are moved along the track pushed by electrical currents. The information is read and written by a write head and read head with fixed positions. With respect to a memory implemented with fixed cells, the racetrack memory implementation has advantages in the number of writing and reading devices required for the device function and in the maximum density achievable. The disadvantage is mainly in the latency of reading and writes operations. The designed LIM circuit is based on the racetrack structure and it has been developed starting from the basic principle that the LIM circuit should work also as a common memory as long as the operation is not required. The logic has been integrated, taking advantage of lateral tracks to connect the skyrmion logic, as already demonstrated in the previous chapters. In addition, the lateral track is gated with a VCMA gate. The skyrmion stored for reading and writing purposes should indeed be free to move in both directions along the racetrack without any risk of information loss. The logic has been integrated, taking advantage of lateral tracks to connect the skyrmion logic, as already demonstrated in the previous chapters. In addition, the lateral track is gated with a VCMA gate. The skyrmion stored for reading and writing purposes should indeed be free to move in both directions along the racetrack without any risk of information loss. The next step in designing a LIM device is the evaluation of the storage position of the elaborated information in the array. This choice impacts heavily on the whole circuit's final performance and is strongly dependent on the chosen algorithm and its implementation. Later in the chapter, section 6.2.5, it will be shown how the information is moved during the algorithm, especially in MixColumn operation. In general, for the AES algorithm, the elaboration requires a great number of move operations repeated at every round in the algorithm. Therefore, the information placement is the most impacting choice in terms of time required by the algorithm and, consequently, the operation's overall energy. Minimize the distance that skyrmions have to travel to reach their final destination for processing is fundamental to obtain good performance.

The information is organized as shown in figure 6.8. As shown, the information storage position recalls the state structure used in the matrix representation 6.8. The difference with that representation is that the elements are stored in a transposed way, the row elements are placed as shown in order along the vertical tracks.

In the implemented algorithm all the shift and communication operations are executed between the elements belonging to the same row and column. The movement with the highest number of repetitions is the one between elements of the same column in the state matrix during the MixColumn operation, due to the matrix multiplication described in section 6.1.5. In the proposed mapping, the movement, required during the sum of partial products, will require only a single movement operation per position. On the contrary, a movement along the row of the state matrix, e.g from element W21 to element W23 will require 16 cycles, equivalent to 8 movements per position.

### 6.2.3 Memory Cells interconnect

The movements required by the algorithm are the ones along the memory tracks and among them. To enable the connection between cells belonging to different tracks, orthogonal paths have been added to the array. Following this path a word can be moved to the next vertical track. This connection, equivalent to a shift along the column of the reference state matrix, will be used in MixColumns operation for the partial product sum. Regarding the vertical movement, an additional track is placed side to side to the main racetrack memory to avoid collisions during the movement operations. In rotation operations indeed not all the information on

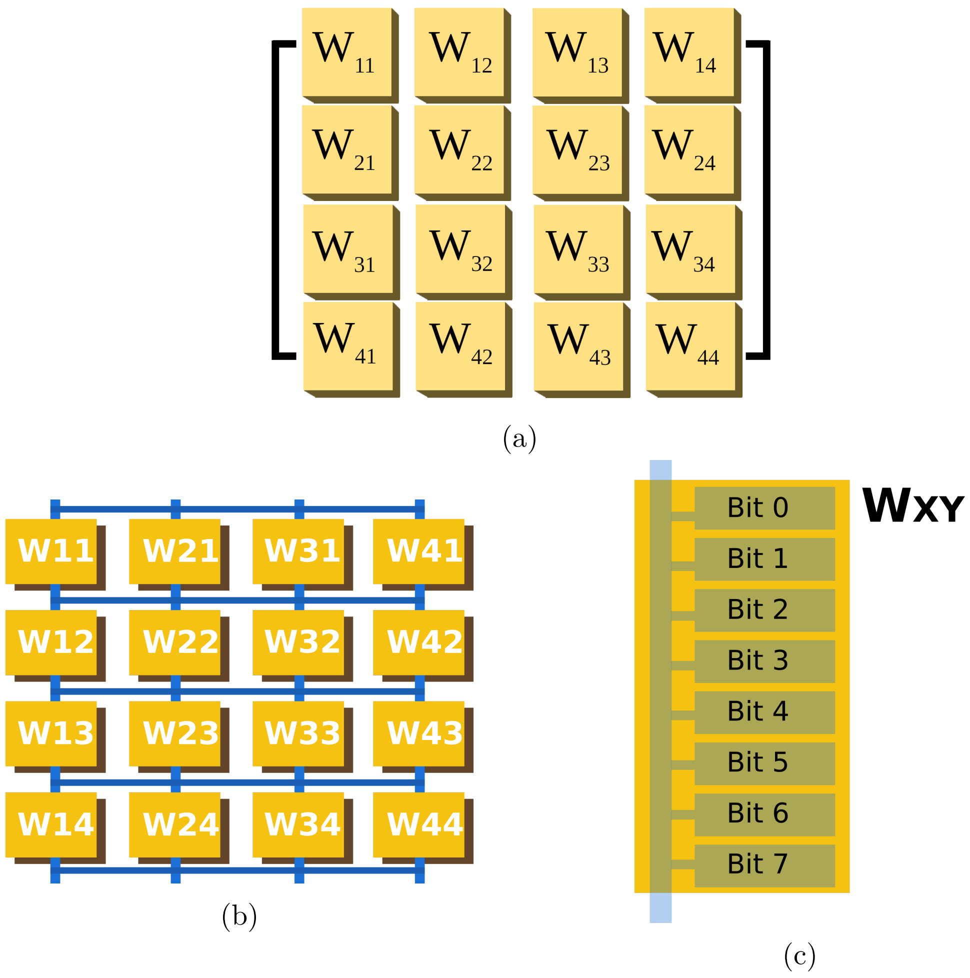
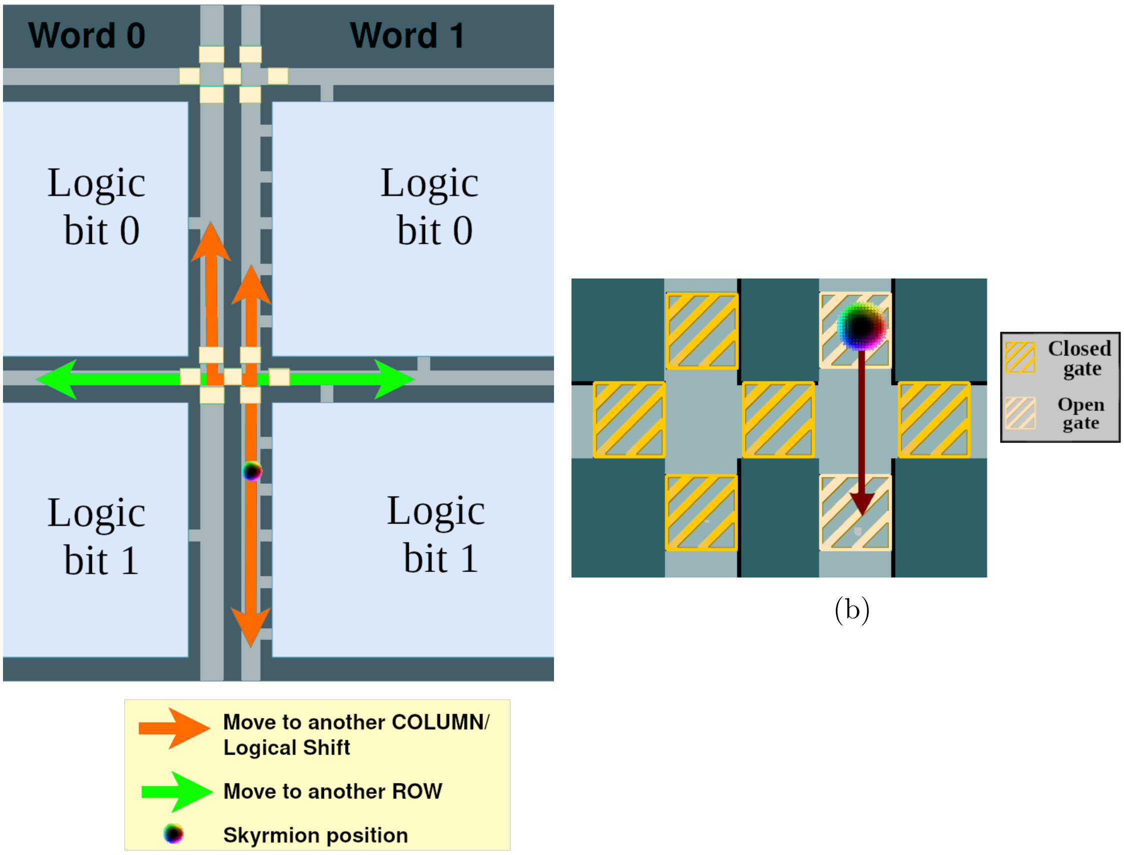


Figure 6.8: Word organization in the AES LIM array with respect to the reference state matrix. a) Reference state matrix b) LIM array mapping c) Bit position inside a single word

the same vertical track moves in the same direction. This requires to maintain still some cells while the other moves and viceversa. In this particular situation, that is common during Shift Row operation, the additional vertical track acts as a side lane in which moving skyrmion can safely pass without affecting the stored information. This additional track is also the track used to collect the results of logic operations. An example of the shift row operation is shown in section 6.2.5.

With the described connections the final track structure of two neighbor words is shown in figure 6.9a. At the crossing between the tracks some additional gates are needed to impose the correct direction to the skyrmions. For this reason a cross matrix has been designed. The structure is shown in figure 6.9b. As shown different VCMA gates are placed along the structure to correctly guide the movement of the skyrmion going through avoiding any deviation due to the SkHE. The control of these gates is electrical. The signals used to control a single cross matrix can be shared by different cross matrices in the array depending on the direction required. In figure a simple example is shown, during the first write operation the information needs to travel from top to bottom. Due to the skyrmion hall effect without any external intervention the information would deviate from the intended direction. With the control added, it is possible to make the information travel along the track. In the same way changing the VCMA gates status it is possible to guide the skyrmions in another direction.



(a)

(b)

Figure 6.9: Interconnection scheme of two neighbor words in the LIM array. (a) Crossing matrix used for AES LIM implementation. The yellow zones corresponds to VCMA gates used to control skyrmion direction

### 6.2.4 Memory Cell Design

The next step in the design of the LIM device is the logic implemented by a single cell.

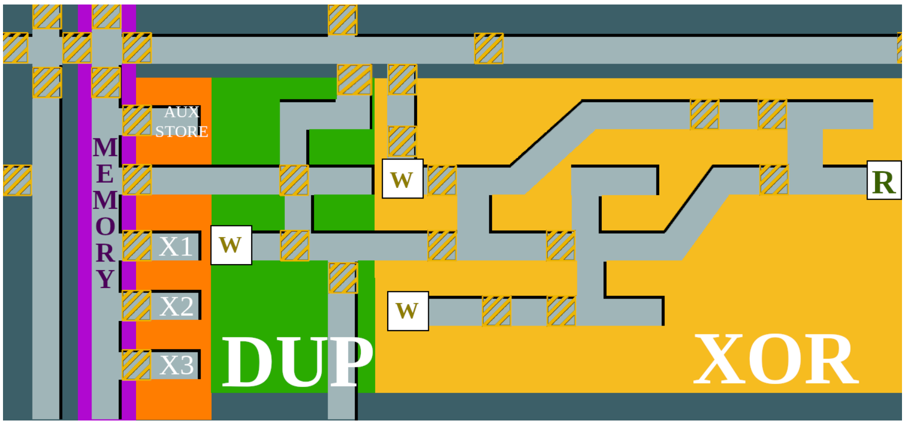


Figure 6.10: AES memory cell. The highlighted squares are VCMA gates

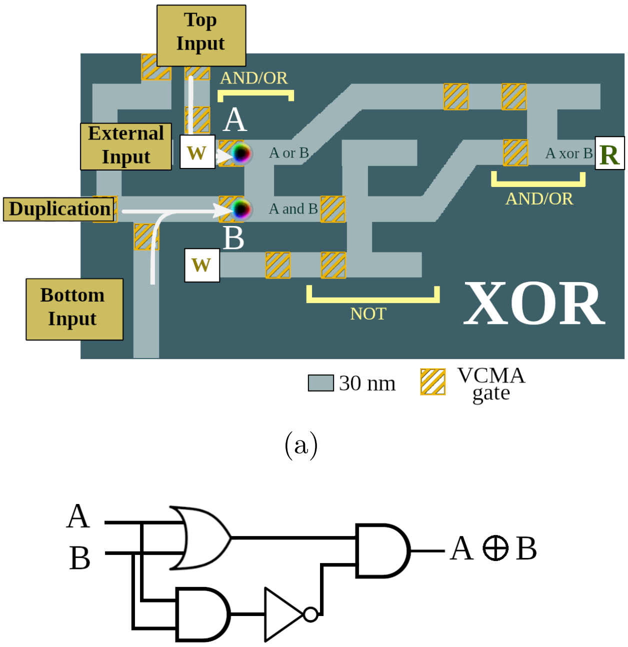
The cell has been designed taking into account the information movements and the logical operations required by the algorithm. The logical operations required are two:

- XOR operations used in Add Round Key, Mix Column
- Shift operations used in all the logic operations

With shift operation, it is referred any kind of operation in which a bit of information is moved from its original storage position inside the array to another. This comprehends logical shifts of single words and movement of entire words from their original position. When a logical shift is needed a current in vertical direction is pushed and the skyrmion will move accordingly reaching other cells. Simple logical shifts can be obtained in this way.

Rotation operations are executed through the interconnection structure described above. Rotations are executed moving the information along the memory track and taking advantage of the additional track described in the previous section.

To perform the algorithm, every cell needs to be able to perform the XOR operation. This operation is used in 2 of the main rounds: AddRoundKey and MixColumns. The XOR operation is performed by means of the gate shown in figure 6.11a. The gate realizes the XOR operation following the logical scheme presented at the bottom of figure 6.11b. The gates are realized using the basic gates presented in section 2.2.2. The synchronization of information is obtained by means of VCMA gates. The inputs of the operation are sent to the first AND/OR gate. The AND output, the line that continue on the bottom is negated by means of



(a)

(b)

Figure 6.11: XOR logic implemented in the LIM Cell. The information for the A and B input of the input can be produced from different sources. The white box indicate external write heads for input generation

a NOT gate.The OR output, the line on top, continues in direction of the last gate. Finally the two results are elaborated together to produce the XOR result as the result of the final AND operation. The result can now be collected in the vertical track To keep the inputs synchronized, the output of the first OR operation on the top track of the first gate is delayed by one cycle by means of an additional VCMA gate. Finally for the correct function, the gate needs one additional skyrmion for the NOT operation. This skyrmion needs to be generated at the bottom input of the not gate, every time a XOR operation is required.

The gate can receives inputs from four different sources:

- Directly from the memory cell after a duplication gate.
- From the top connection track
- From the bottom connection track
- From a write head placed inside the cell

These inputs are indicated in figure 6.11a. The inputs coming from the memory have to go through a duplication gate in order to ensure the initial information is not lost. The gate serves also the purpose to maintain intermediate results over different operations when they need to be reused for later computations. The duplication process is executed taking advantage of the NOT/COPY gate presented in 2.2.2, using the top and bottom exits that produce a copy of the input information. This process needs an additional skyrmion that will produce on the bottom output the copy of the original information. The input from the top and the bottom tracks are used when the information to be elaborated do not belong to the memory cell or in case the duplication has to be avoided.

The input coming from the write head is used to provide external inputs to the computation. In case a 1 is needed as input, a skyrmion is nucleated in that position and can be sent to the input.

Finally as shown in figure 6.10 the main memory track has 4 lateral pockets that serves as additional storage positions. These positions are designed to keep the information safe while the processing is happening. In fact for some matrix operations all the partial products are computed in a first phase and then summed later. In this scenario the partial products are saved inside the cell without any information loss. The cell with this improvement can store more than one information at the same time in a cell. The save and load mechanism is briefly showed in figure 6.12.

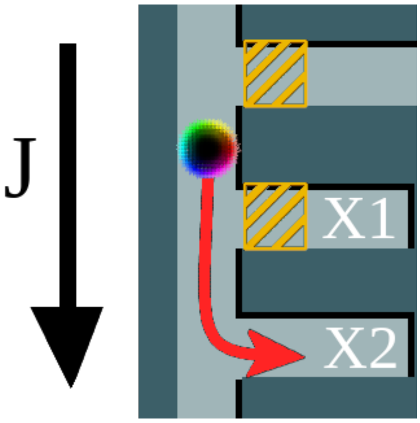


Figure 6.12: Insertion of information in auxiliary storage position. The auxiliary position is used as temporary storage for partial computations.

When the auxiliary storage is needed the VCMA gate is opened and the information can be stored inside waiting to be processed. When the information is needed the correspondent gate is opened and the value is restored. In addition the bottom auxiliary storage is dedicated for the original information. Storing the original information in this position the operation can be safely aborted without any data loss.

### 6.2.5 Control Logic

To control the correct execution of the algorithm a control logic is needed. In particular this logic has been realized in CMOS technology. This technology was considered the most suitable to realize the control of the device. Apart from the clear advantage in design, given by the well-established CMOS toolchain, the electrical nature of control signals gives an important advantage in distributing the control over a wide area in an acceptable time. A single control unit can in principle control a huge number of equal AES-LIM units. In this last scenario the control overhead would be shared by many units reducing notably the impact on performance of the control unit.
To realize the complete algorithm the control logic is organized in a hierarchical way.

#### Master FSM

The complete operation is guided by a Master FSM that deals with the four main operations and the rounds. The master FSM synchronizes the operation of 4 slave FSM dealing with the control of the single operations. A single slave FSM is dedicated to every main operation except for AddRoundKey and SubBytes, that have been joined in the same FSM. The fourth slave FSM deals with the Key expansion to produce the different RoundKeys used for the AddRoundKey operations. The figure 6.13 shows the master FSM. The single control signals are not reported to maintain the figure as clear as possible.

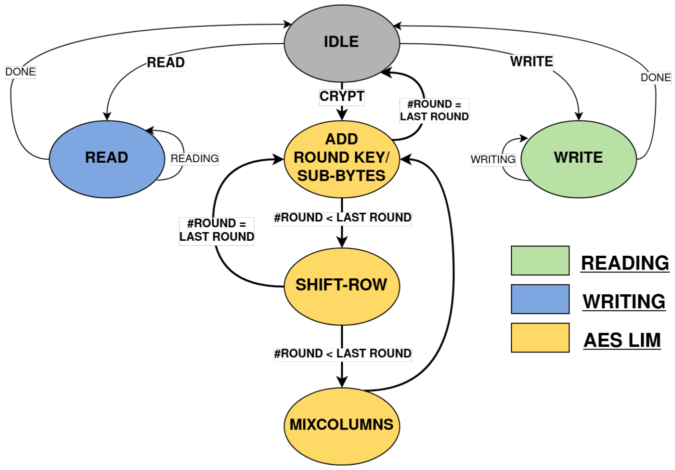


Figure 6.13: Master FSM states diagram. The FSM manage AES, write and read operations

The read and write operations are managed by the same state machine.

### AddRoundKey - SubBytes FSM

Following the encryption path of the Master FSM the first state deals with the AddRoundKey and SubBytes operation. When the master FSM reach this state a trigger is produced for the corresponded slave FSM. The FSM, shown in figure 6.14, guides the information in the AddRoundKey and Subbytes operation:

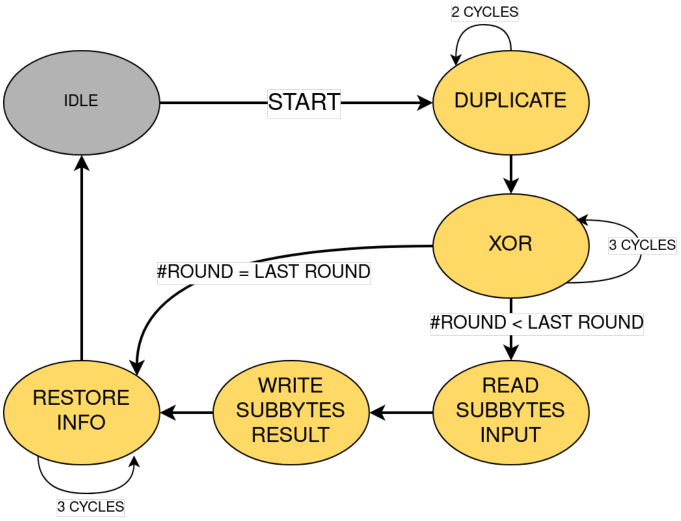


Figure 6.14: Add round key and sub-bytes FSM states diagram. The FSM manage AES, write and read operations

The states shown represent the main operations executed during the Add Round Key and Sub Bytes process. Every operation corresponds to a special function of the LIM array that can require more than one clock cycle to complete. The presented slave FSM is triggered at the beginning of the algorithm and after every Mix Columns operation. As can be seen from the schema in figure 6.14, the execution flow is linear and it is altered only in the last round where the Add round key ends the complete algorithm without the need for a SubBytes operation as previously shown in figure 6.2. The machine follows the following basic steps:

- *Duplicate* The information initially stored into the cell is duplicated. One copy of information can now be processed safely into the XOR circuit inside. During this operation the round key bit is nucleated inside the cell.

- *XOR* The xor operation between the original information and the round key bit is executed as discussed in section 6.2.4. In 3 cycles the results is ready at the end of the cell.

- *Read SubBytes Input* The information is read from every cell and sent to the SubBytes LUT for the substitution.

- *Write SubBytes Result* The SubBytes output is sent back. Now, if the read bit is equal to the the received information the information is kept. On the contrary if a 0 was read and a 1 was obtained by the SubBytes LIM a new skyrmion is nucleated at the write head present inside the cell. Viceversa, if a 1 was read the information is simply annihilated.
- *Restore Info* The information should be sent back to the store position in the main track. The cross matrix is then correctly set and in 2 cycles the information reaches the storing position

As already explained, in the last round the *Read SubBytes Input* and *Write SubBytes Result* are not executed. The information at the output of the XOR gate is then sent back to the storing position.

### Shift Rows FSM

The next step in the algorithm is the ShiftRow operation. As explained in section 6.1.4, the information has to be shifted by a fixed amount depending on the row the element belongs to. The information positioning in the memory array is transposed with respect to the original matrix representation of the state, therefore the shift row operation corresponds to a vertical movement of information. The slave FSM is one of the simplest. The only operations it has to complete are an up movement and a down movement. The FSM signals are sent only the cells that are subjected to the rotation operation. The FSM is shown in figure 6.15. The move up states are kept stable for 24 cycles. In the worst case indeed the words needs three times the cycles for a word shift to reach the correct positioning. The same is valid for the down movement.

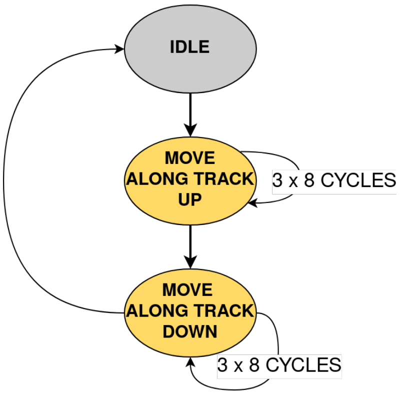


Figure 6.15: Shift row FSM states diagram.

The additional track serves its purpose in this operation of the algorithm. In fact the information is deviated into the backup lane to avoid any collision with the words that are kept still during operation as shown in figure 6.16. After the first up movement the information is now shifted to its final position on the backup lane. The remaining information is then shifted in the other direction taking advantage of the last cycle to move also the information in the backup lane in the memory track.

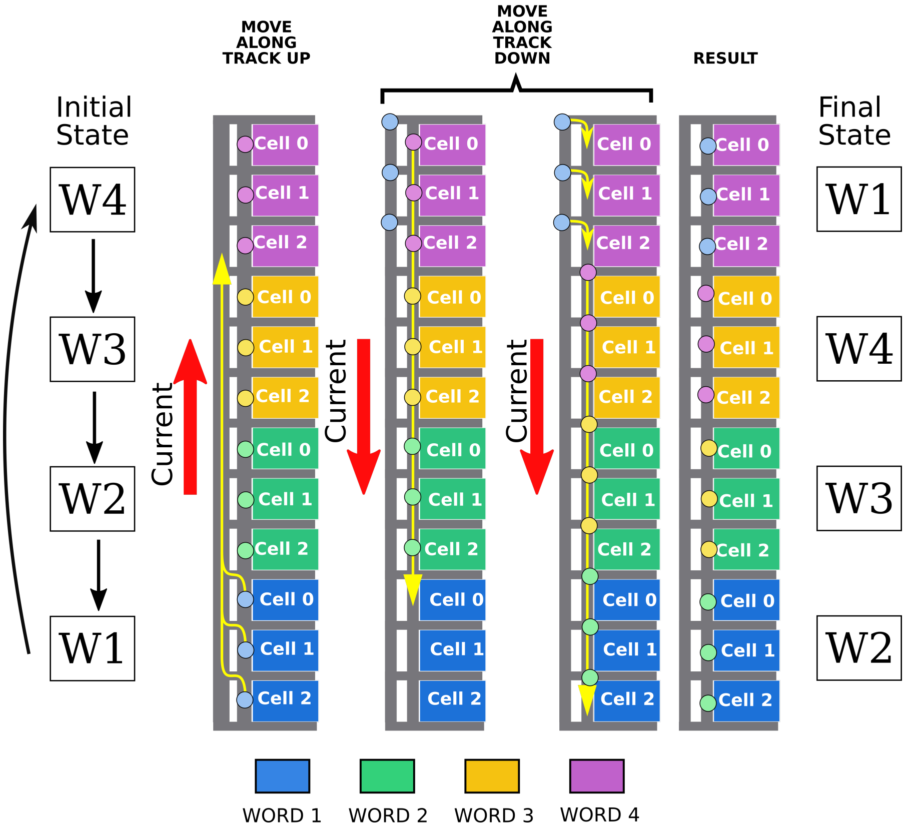


Figure 6.16: ShiftRow operation example. The rotation moves the word 1 in position of word 4 and shifts the other 3 words. The example reports 3 bits words, the algorithm is executed on 8 bits word.

All the rows involved in the rotation process are moved at the same time. The different movement is realized with different configurations of the the cross matrix. With this strategy the time required for operation is equal to a single iteration of the FSM.

### Mix Columns FSM

The FSM that controls the array in the mixColumns operation is the one showed in figure 6.17. The operations can be virtually divided in two sets:

- Computation of the partial products

- Sum of the partial products to compute the new state matrix

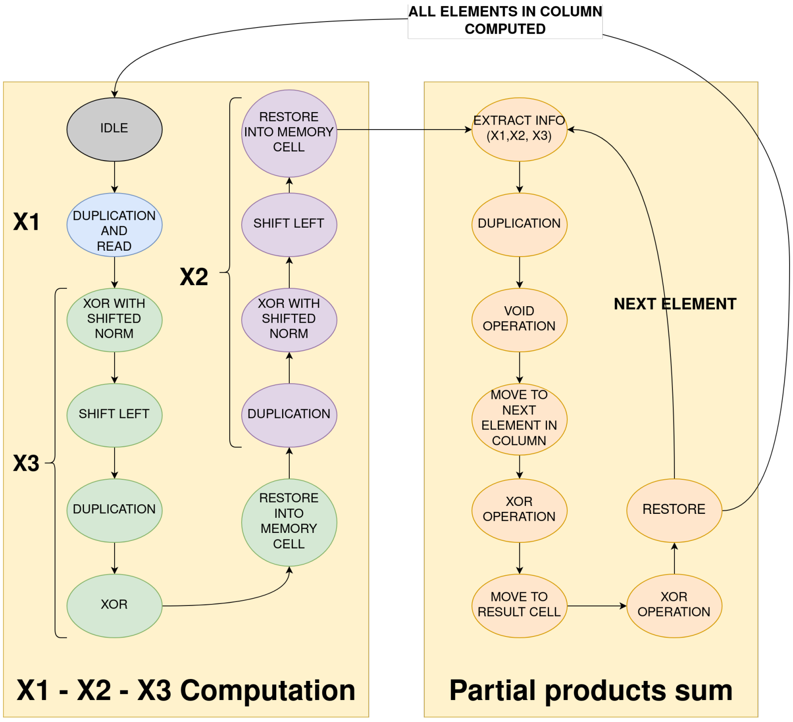


Figure 6.17: MixColumns FSM state diagram

The matrix used for the mixColumns operation, showed in section 6.1.5, is constant for every round. Therefore, the execution of this state machine is equal at every new iteration. The only difference between rounds is the step *XOR with shifted NORM* in which the word term is XORed with the normalization term. This normalization term is used as explained in section 6.1.5 only when the MSB of the shifted word is 1. In case a 0 was read in the MSB position, the normalization term is substituted by a word composed by all 0s, neutral element of the XOR application.

To execute the MixColumn result, at first the results of multiplication by 1, 3 and 2 are computed in order. After every computation the result is stored in dedicated zones into the memory cell. These zones will store the partial results that will be reused later in the algorithm.

The main operations of the partial products computation are:

- *Duplication and read*: all the bits of the words stored in the array are sent to the duplication gate except for the MSB of every word that is read. The result of the read operation will decide if a normalization is required. The duplication produces the first partial product the one multiplied by 1.

- *XOR with shifted NORM*: depending on the value read at the MSB in the duplication and read state the normalization factor of Galois field multiplication is summed with the content of the memory. The normal order of operation in the AES algorithm is the opposite but executing at first the normalization and later the shift allows to spare cycles in the operation.

- *Shift Left*: the result of the previous operation is directly sent in the next cell as top input for the XOR operation.

- *Duplication*: the original data (X1) is now duplicated. One copy will be the input of the XOR operation together with the shifted result of the previous XOR. The other copy will be restored in the cell.

- *XOR*: the XOR to compute the X3 partial product is finally executed.

- *Restore in memory cell*: The X3 result is sent back to the cell and stored in the correspondent auxiliary store position in the cell.

- *Repeat the operation to compute X2*: the four operations that follows allow to produce again X2 that now is sent in the memory cell and stored in the correspondent auxiliary storage position.

Thanks to the auxiliary position now every cell in the LIM array contains a X1,X2 and X3 partial products. The sum that will be executed now will take advantage of the associative property of sum to speedup the computation as shown in figure 6.18a and 6.18b. The results are then sent to the two inputs of the cell that will then store the result and then will store it.

The main steps of this second phase are:

- *Extract info*: the gate of the auxiliary storage needed for the computation is opened and the information can now be placed in the cell.

- *Duplication*: the duplication operation is now executed. The execution of this operation is needed only the first time a X1 data is used.

- *Void operation*: the duplication operation is now executed.

- *Move to the next element*: the information now at the exit of the gate is sent to the next corresponding cell.

- *XOR operation*: the first XOR is now executed

- *Move to result cell*: the results are moved to the cell that will store the result. 6.18c

- *XOR*: the result is finally computed.

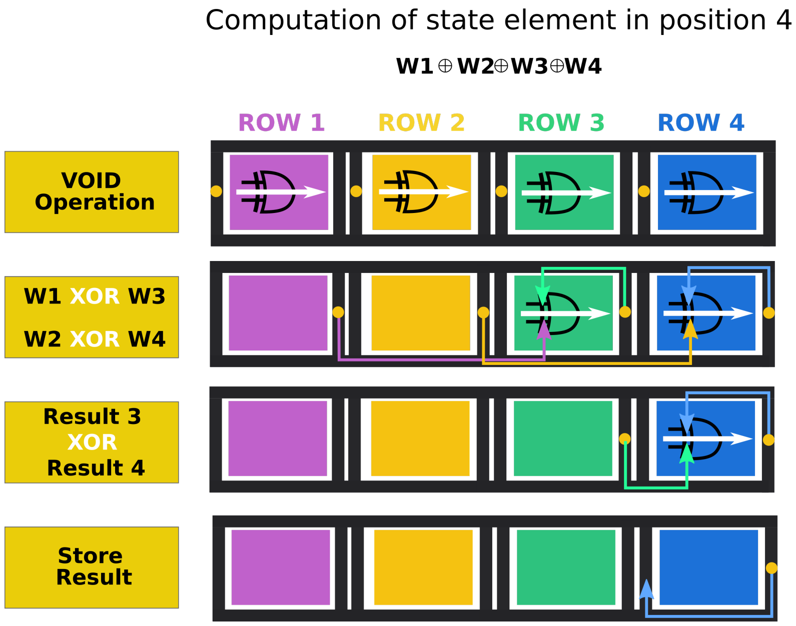


Figure 6.18: Partial products operation example. The target cell in this example is the cell 4. The row indication refers to the correspondent position in the reference state matrix.

- *Restore*: the result is restored into the cell in the auxiliary storage position in which previously the information was stored.
- *Repeat the second phase other three times*: the operation is repeated 3 times until the new state is ready and stored in memory.

The four iterations of this second phases varies only for the selection of the target cell. The FSM is aware of the iteration and change both the current directions and the cross matrix configuration required for movement at every new iteration to be sure the skyrmions move to the right cells.

### Key Expansion Logic

The key expansion logic is completely realized in CMOS technology. The block receives the external key in input and produces all the round keys for the LIM devices following the algorithm explained in 6.1.1.

The computation of a new round key is triggered at every mixColumn operation. The round key is then sent from the round key block to the write control to be then written at the input of the block during the add round key operation. The reason for the realization of the key expansion block in CMOS logic is that the signal produced by the key expansion block should be distributed to many cells at the same time and this data also for security reasons does not need to be memorized for a long time. All the logic linked to the key is indeed volatile.

### 6.2.6 Current generation

The current generation is a critical aspect in spintronic devices. For this design the regularity of the design and the possibility to limit the movement of skyrmions with VCMA gates allows to create a simplified contact scheme as shown in figure 6.19.

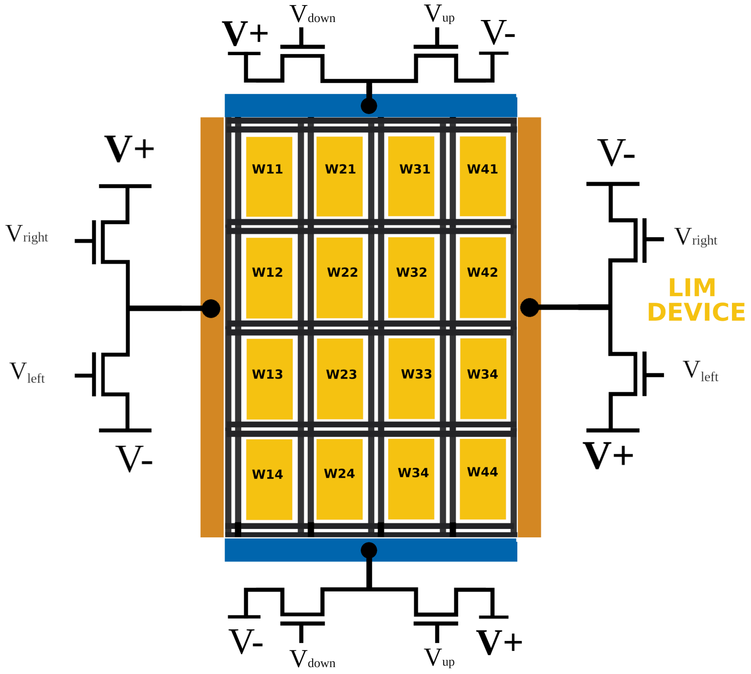


Figure 6.19: Contact scheme for the current generation of the AES LIM device

All the cells are guided by two set of contacts that allows the generation of current in horizontal and vertical direction. This two current direction are used to generate all the movements in the circuit. This schema can be used thanks to the confinement chose for the implementation. The confinement realized with curbed material, in fact, allows to have a uniform current in the array. On the contrary in case of etched structures, the density of the current would have been strictly linked to the skyrmion paths not allowing an even distribution and requiring multiple contacts to obtain the desired movement.

### 6.2.7 Integration with skyrmion memories

The proposed LIM circuit is based on a racetrack memory. There is then a preferable way to communicate to other racetrack memory devices: taking advantage of the information movement already used in the ShiftRow and MixColumn operations. The advantage, connecting the LIM array to additional tracks, is the additional high-density storage realized on the same device of the LIM. The additional logic in the LIM memory cells in fact reduces the storage density of the device. In addition the number of LIM devices in a complex memory is limited by the power and energy consumption the device needs to operate. As shown in figure 6.20 the communication with an higher density memory can be realized directly connecting the word lines to the racetrack memories. The connection can be made on the horizontal line or on the vertical line following a serial approach. In case of vertical insertion of information from the memory, the movement needs only an adjustment in the synchronization to match the distances between the cell in the LIM array. In fact following this approach, the synchronization speed has to be slowed down to match the lower density of the LIM circuit with respect to the racetrack memory. Differently, in case of horizontal insertion, the movement of information inside the LIM array needs to take into account the fact that the skyrmions will be stored in the vertical direction. For this reason the control needs to count the clock cycles and update the cross matrix at every new word to correctly position the information. The usage of an additional vertical current is required to shift the skyrmions in the correct position.

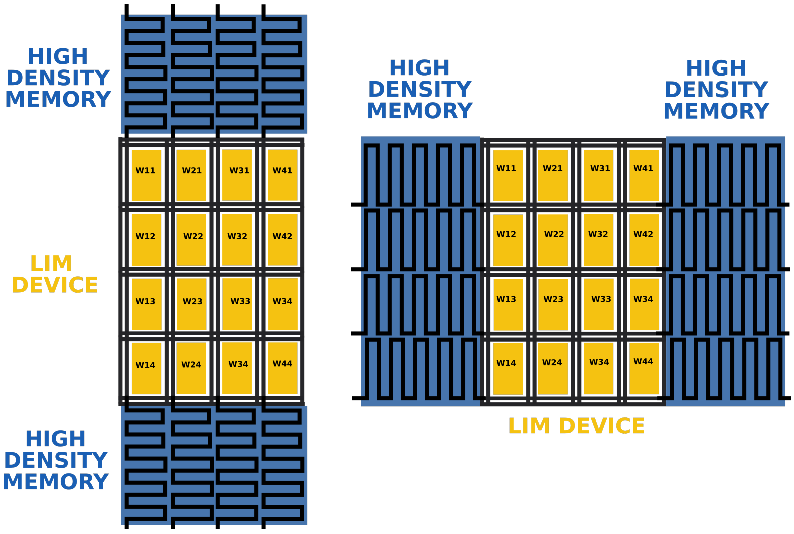


Figure 6.20: Vertical and horizontal topology for LIM device integration with skyrmion racetrack memory

## 6.3 Simulations and performance evaluation

### 6.3.1 Methods

The performance of the array has been evaluated with micromagnetic simulations of the memory cell shown in figure 6.10. The memory cell was simulated with a curbed track structure and the parameters shown in table 6.1. This parameters refers to W/CoFeB/MgO thin film.

Table 6.1: Parameters used for micromagnetic simulations and current distribution computation

| SIMULATION PARAMETERS | | |
|---|---|---|
| Saturation Magnetization [52] | 1e6 | $\mathrm{A\,m^{-1}}$ |
| Uniaxial Anisotropy Constant [52] | 8e5 | $\mathrm{J\,m^{-2}}$ |
| Exchange Stiffness [52] | 2e-11 | |
| Damping constant [75] | 0.03 | |
| Spin Hall Angle [63] | 0.4 | |
| Film resistivity [63] | 165 | µΩ cm |
| Curb width | 30 | nm |
| Curb depth | 0.5 | nm |

Finally regarding the power and energy consumption required for a complete operation the contact schema explained in section 6.2.6 has been used. Except for the duplication stage the movement of the skyrmion is generated by the contacts at the periphery of the circuit, as shown in section 6.2.6 The power consumption has been extracted considering the current flowing in the whole array and considering the resistivity of a W/CoFeB/MgO stack [63].

### 6.3.2 Performance evaluation

The first parameter that has been evaluated regarding the performance of the circuit is the minimum execution time. In such architecture the critical path is given by the longest path the skyrmion has to travel. As shown in figure 6.21, the cell longest path is represented by the horizontal movement. This movement is used in the partial product sum phase for the MixColumnOperation and every time the result of the XOR operation has to be restored in the memory track.The time required for a complete shift with a current of magnitude $2.67 \times 10^{10}\,\mathrm{A\,m^{-2}}$ is 5 ns. The execution time of the algorithm can be computed as the number of cycles required for the algorithm multiplied by the time of the critical path, assuming a regular movement of the skyrmions. The execution time in this scenario is 5.88 µs. Allowing to the circuit to reach, at maximum speed, a throughput of 18.1 Mbit/s. The throughput refers to a single LIM unit. This value in real implementations should be computed knowing the number of LIM devices employed in the circuit. This number depends mainly on the cost/performance balance. The mean power required by the presented implementation at the maximum tested speed is 245 µW. For reference it was considered the power required by a low power CMOS AES accelerator like the one presented in [90]. The power has a value of 4.39 mW. The LIM with the same power can run 18 complete LIM units. Therefore the throughput of the complete device rises to 392 Mbit/s compared to the 439 Mbit/s of this reference low power implementation. Despite the value of the accelerator remains still higher than the proposed implementation, it is worth to consider that the reported power of the accelerator does not take into account the cost of data movement that is absent in our LIM implementation or limited as explained in section 6.2.7. Especially in case of encryption of large data, this movement can be really impacting, considering that AES works on 128 bit blocks. An example of this has been presented in [91], where the authors presented an in-memory implementation based on phase change memories and STT MRAM and compared with a dedicated accelerator sitting out of the memory for AES encryption. It was showed in this case that for single 128 bit acceleration the proposed CMOS dedicated solutions were better even considering the movement of data. On the opposite when a large amount of data were processed, a case of study of 1Gbit was considered, the CMOS based accelerator was outperformed by the in-memory implementation, especially in the case of parallel units for computation. For the presented studies no further comparisons were done about CMOS implementations as they would have required further evaluation of the skyrmion memory system, encryption block strategy and surrounding circuitry that is out of the scope of the presented study.

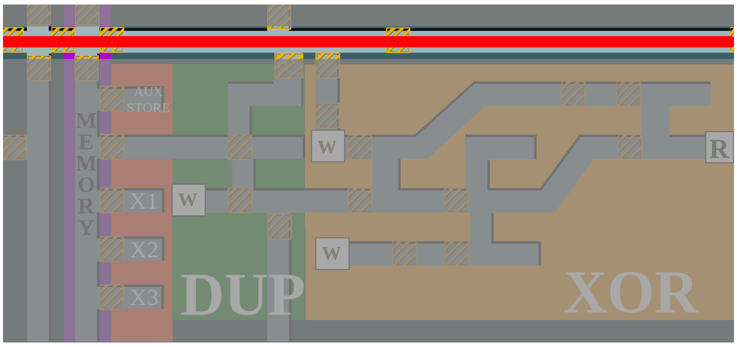


Figure 6.21: Critical path through the basic LIM cell

The performance of the proposed architecture have been compared also with other LIM implementations based on different technologies as, CMOS-nanowire-molecular (CMOL) and domain wall motion [92]. This comparison is provided in order to give a reference to the proposed architecture with respect to other devices implementing the AES encryption algorithm in memory. The performance are reported in table 6.2. The working frequency of the circuits is fixed at 30 MHz. For this reason the performance of the proposed skyrmion solution have been evaluated considering a current of $2.53 \times 10^{10}\,\mathrm{A\,m^{-2}}$ and a cycle time of 6 ns. The current was chosen near the minimum value that is necessary to obtain a working architecture. In this condition a pulse time of 6 ns is enough to move the skyrmion in the circuit. For lower current especially the logic functions realized by the XOR and duplication part of the gate do not behave in the desired way leading to skyrmions pinned in the junctions between tracks.

Table 6.2: Comparison of AES LIM implementations

| | **Cycles** | **Power**<br>[µW] | **Energy**<br>[nJ] | **Efficiency**<br>Gbps/W | **Area**<br>µm$^2$ |
|---|---|---|---|---|---|
| SHE-DDWM-1 [93] | 4176 | 11.49 | 1.6 | 80.02 | 127 |
| SHE-DDWM-2 [93] | 2168 | 24.07 | 1.74 | 73.58 | 272 |
| SHE-DDWM-3 [93] | 1084 | 53.13 | 1.92 | 66.67 | 508 |
| CMOL [94] | 470 | 657 | 10.3 | 12.43 | 320 |
| Baseline DW [95] | 1022 | 57.5 | 2.4 | 65.345 | 78 |
| Pipelined DW [95] | 2652 | 26 | 2.3 | 55.69 | 83 |
| Multi-issue DW [95] | 1329 | 61 | 2.7 | 47.36 | 155 |
| Proposed | 1386 | 50.1 | 2.6 | 47.8 | 53.3 |

The proposed solution showed a good power performance compared to the other solutions. The power of the proposed solution indeed is comparable to the some of the other solutions even the consumption remains high compared to some of the optimized architectures. The performance of the proposed solution are limited by the minimal current required for the correct function of the logic, indeed the employment of a lower current density allows to reduce with a quadratic relation the power required by the circuit. For this purpose a further optimization of the logic would be needed to allow to the skyrmion logic to operate at lower currents. The energy of the operations was then computed considering the execution time and the mean power consumption. The expression used for the computation is $E = P_{mean} * t_{execution}$. The value obtained considering a single LIM array is 2.6 nJ. This value, that takes into account all the moving operations, read and write and the VCMA gate requirements is higher compared to many of the other solutions. This, in a similar way to the power consumption, depends in the most part on the current density applied to the circuit. The implemented solution cannot perform easily under the value of $4 \times 10^{10}\,\mathrm{A\,m^{-2}}$. The most critical part in this sense is the NOT/COPY logic gate where the movement of the skyrmion at current lower than $4 \times 10^{10}\,\mathrm{A\,m^{-2}}$ are more sensitive to small deviations in the position of the input skyrmion, resulting in some cases in wrong results. In this implementation, differently from the previous, the logic performs without the initial pulse of

$20 \times 10^{10}\,\mathrm{A\,m^{-2}}$. This to maintain the values of energy and power low in these stages that are not critical for the timing of the device. Finally, the efficiency of the architectures was computed. This efficiency was computed as the ratio between the throughput that can be obtained by the architectures and the power required to the device to operate. The computed value is equal to 47.8 Gpbs/W. This value is important in this kind of implementation because, given a power budget for a particular circuit, it sets the maximum achievable performance. It is important to underline that the parameters have been computed neglecting the cost of the control. This hypothesis is acceptable in case of LIM architecture because is often possible to guide a complete set of LIM circuits with a single control. In this situation, the power and energy required by the control is shared among all the units involved in the calculation generating then a negligible effect on the overall power and energy consumption of the circuit. Regarding the comparison, the other implementation did not include the cost of an eventual control as well. The comparison is therefore fair between the different technological solutions. Finally some considerations should be done on reading and writing operations performed during the algorithm. The number of reading required during operation is limited, indeed the read of complete words happens only in the SubBytes phase to realize the substitution of information after the AddRoundKey operation. In addition to that, at every MixColumn operation, the MSB of the 16 words composing the actual state is read another time to know if the word needs normalization for the shift operation as explained in section 6.1.5. Summing all, the read operations in a complete encryption are 1424. The energy required by a reading operation was estimated in 91 fJ [96]. Therefore, the total energy required for read operations amounts to 136 pJ. In the energetic evaluation also the energy to switch on and off the VCMA gates for synchronization was considered. The energy to activate the gate has been computed to be 0.45 fJ per charge. The total energy to complete the algorithm was estimated in 88 pJ. The writing operations are in the proposed architectures critical for the execution of the function. The write operations are used in many phases of the algorithm. In particular a write operation is performed during the duplication operation, when an external input, like the round key, is used, when the substitution word is written in the SubBytes process and for the auxiliary skyrmion needed for the XOR operations. While the skyrmion generated for the duplication and the auxiliary are mandatory for the correct function operation, the number of skyrmions written by the external inputs can vary depending on the number of ones required during the computation. Considering the worst case scenario in which all the round keys and the SubBytes operations require a write, the number of nucleations of skyrmions is equal to 23424. The energy required for the writing operation has been computed considering a writing operation by means of VCMA of 6 fJ as computed in [76] for enhanced VCMA coefficient materials . The value obtained is 140 pJ.

The last parameter of the comparison is the area. The area was compared only with LIM implementations because in that value is also present the space occupied by the memory. In case of common accelerators this value is not present and its sum to the memory system is not trivial due to many different variants the memory system can have. In comparison with the other LIM systems the skyrmion implementation occupies a small area equal to 53.3 µm$^2$. The implementation indeed, as already shown in the minimum maximum implementation, section 5.7, benefits from the small dimensions of skyrmions and the small range of their interactions that allows to maintain the gates very compact. Compared to the other implementations, as presented in the case of CAM memories, the density achievable from such devices is remarkable. This is a strong point for the adoption of skyrmion logic in memory technology compared to other solutions present in literature.

Finally due to the addition of logic to the circuit, a reduction in the density of the information stored in memory was expected. This parameter should be taken into account, being the circuit also designed as a functioning memory. In particular in the LIM array the memory linear density that in a skyrmion racetrack memory based on the same structure is 16.6 bit/µm is reduced to 2.3 bit/µm. This factor affects linearly the reading and writing latency in the circuit. The reading and writing mechanism employed in racetrack memories is serial, therefore a lower density will result in longer distances to travel and consequently in lower read and write speeds [97].

### 6.3.3 Final considerations

In conclusion it was demonstrated a possible design for a LIM device capable of executing AES encryption based on skyrmion technology. The device showed acceptable performance compared to other similar spintronic implementations with comparable or slightly worse values in terms of energy and power. In addition, similarly to what is was showed in the implementation of other algorithms, the area occupied from the architecture proved to be the best of the LIM implementations taken for comparison. Such characteristic is important in logic-in-memory implementations. In this scenario the best performance are achieved when the inherent parallelism of logic in memory is exploited. In addition the small area allows to not impact too heavily on the memory density. These devices in fact are more effective as computing solution if they can serve also as main memories in computing systems it is therefore important to achieve the highest possible capacity in order to reduce at minimum the data movements required. At the same time the architecture requires many VCMA gates to correctly control the movement and the requirements of many write operations and can make the implementation of such circuit challenging. Finally the energetic performance can be further optimized considering an optimization of the structure to work at lower values of current density. A lower value of current corresponds to a lower value in the power required by the circuit to realize it's functionality and this can eventually change also the energetic consumption of the implementation. These performance are important in a LIM implementation where many different units are expected to operate in parallel. Another characteristics that deserves further attention in future is the confining mechanism. The small feature dimensions required for the curb in which the skyrmion travels, makes the implementation of this structure in a real device not trivial. Therefore, alternatives for the confinement like anisotropy gradients needs be explored in future, in order to reduce the challenges in the production of such devices.

# Chapter 7

# Skyrmion LIM: Final considerations and future works

In this section some considerations on the logic-in-memory implementations with skyrmions are discussed. As shown in the previous sections the skyrmion technology, can be used successfully for logic-in-memory applications. In particular the skyrmion memory can be easily integrated with simple logic like the one usually required by LIM devices. In addition, as shown in both the architectures, the current-induced movement of skyrmions can be used to efficiently move information in the LIM array. Finally, the low power required for the movement allowed to obtain good performance in all the proposed implementations. One of the strongest points showed in all the implementations was the low area occupied by the implemented devices. In particular in case of LIM devices this is a very important characteristic. This architectures benefits from the high parallelism of the implementation and suffer if the data needs to be read and written continuously. [98, 91]. The low area in this scenario allows to keep together an high number of LIM units in the same device. This allows also to have devices with very high capacity having at the same time, that can therefore work as the main memory of a normal system but that can, at the same time, provide the logic functionality without the need of expensive read and write cycles from and to the memory itself. Regarding the energetic performance the technology is competitive with other technologies only in few cases. In particular the energetic cost of the movement is high especially if the system is pushed to reach high skyrmion velocities. In cases in which the system can accept lower execution times like the AES case the performance of the system can be better. Despite the ease of integration and the good performance showed especially in terms of area, complex architectures like the one presented in the AES case enlighten different limitations and challenges linked with skyrmion logic implementations. The first is the duplication of information. As shown in the previous chapters a domain wall pair - skyrmion conversion mechanism was used for this purpose. This mechanism requires high energy densities and careful control of currents with respect to any other movement required for the computation. To achieve this goal, as shown in the Max/Min implementation a dedicated set of contacts was designed. These requirements, added to the possible imperfections in the material, not considered in the simulations performed in the previous chapters, can make the design of such device critical given also the importance this particular gate has in logic operations. An alternative to this gate is the NOT/COPY gate proposed by Chauwin et al in [55]. The NOT/COPY, employed in this thesis for the realization of the duplication in the AES implementation, solves the issues regarding the high current and the complex gate shape required in the conversion from domain wall to skyrmion. At the same time, this gate requires an additional skyrmion for correct operation. This requirement forces the designer to consider a reliable way to nucleate a skyrmion in the auxiliary position or to guide a skyrmion in that position as suggested by Chauwin et al. in their original proposal [55]. In the former case, the circuit requires the placement of many write heads in the circuit, as much as the duplication gates employed in the circuit and the design of a dedicated control to generate a skyrmion only when required. Differently, in the case in which information should be routed to the auxiliary input, a mechanism is required to reliably guide the skyrmions through the structure and be sure that the auxiliary input has always a skyrmion ready for a new operation. This require non-trivial routing of information and external control. Another limiting factor of skyrmion logic is the complex routing of tracks required for cell communication. In particular, as shown in the AES implementation, being limited on a single layer for information movement the tracks crossings are inevitable, especially in architectures where different paths are available for logic operations. The crossing of tracks requires a complex control to obtain the correct direction of information as shown in figure 7.1 in which for the most complex crossing 7 VCMA gates were used. The complexity can be reduced especially in case the skyrmions travel in a single direction in every track but pones two limitations: the materials without the possibility of local control of anisotropy require more complex cross structures to avoid the deviation of skyrmions and the information traveling in the structure should be carefully synchronized to avoid collisions. The previous presented limitations represent great challenges for skyrmion technology, for this reason alternative solutions should be explored in the future. As an example, the possibility to route skyrmion in multiple layers would give to the computation architecture based on skyrmion a great advancement. In particular the possibility to move skyrmions in other layers would allow to greatly simplify the control complexity and to realize easier communication between logic devices across the circuit. In future it will be also important to bring the presented simulations and evaluations closer to a real case considering also the temperature effects the roughness of the materials and other characteristics as the grain structure of the material. In fact the introduction of this factors can be very impacting especially in the logic part of the circuit where the skyrmion has to travel in complex paths and where stochastic movements due to temperature or pinning effects can prevent the correct execution of the logic function. Finally the presented structures has been designed starting from a stack with a curb for skyrmion confinement. Despite the optimal confinement effect, the realization of the presented structure with the required size, i.e. a curb 30 nm wide and 1 nm deep, can be challenging. For this reason an important study in the future will be the application of alternative confinement techniques for skyrmion movement. In particular the confinement realized by material irradiation proposed theoretically by Fook et al. in [73] and recently experimentally demonstrated by Juge et al. in [99]. The local variation of anisotropy in the proposed technique allows a strong confinement of skyrmion that can be compatible with the proposed architectures. Experimentally this technique is much easier to realize and can be a valid substitute for the proposed architectures.

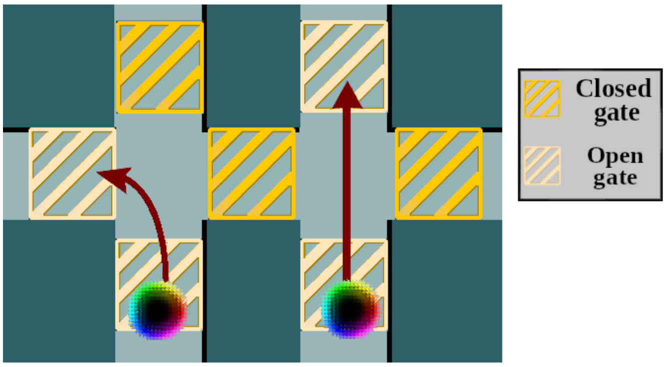


Figure 7.1: Crossing matrix used for AES LIM implementation. The yellow zones corresponds to VCMA gates used to control skyrmion direction

# Part II

# PNML-Skyrmion Interface

# Chapter 8

# Skyrmion-PNML interface

## 8.1 Motivation

The realization of logic-in-memory devices completely in skyrmion technology showed in the previous chapters of the thesis, demonstrated several strong points but also several shortcomings. In particular few of them are particularly impacting on the circuit performance:

- The constraint of routing the circuit on a single layer increases drastically the number of crossing of tracks. Crossings require additional control to correctly direct information and to avoid collisions throughout the circuit.
- Careful synchronization is required with gate based on skyrmion logic due to the small range of interactions between neighbor skyrmions (Section 2.2)
- Duplication is required for correct circuit operation and this operation requires an expensive duplication in DW, as implemented in the previous Minimum-Maximum search implementation. Alternatively as implemented in the AES LIM circuit, the duplication requires the nucleation of a new skyrmion in a logic gate used for duplication.

The presented challenges stimulated the research of alternative approaches in which skyrmions can communicate with other technologies. In 2014 in [33] Zhou et al. discussing their studies on Skyrmion - Domain Wall Pair conversion proposed the idea of using skyrmions as fast magnetic communications means in magnetic circuits. Later in 2015 in [56], Zhang et al. proposed, under another guise, the same concept. In fact, the skyrmion information was translated in a domain wall pair form, elaborated and then translated back into a skyrmion to be moved and used later. The study underlines that thanks to their mutable nature magnetic textures can be manipulated and translated in other forms without losing the encoded information. This manipulation widen the range of possibilities for design of magnetic circuits. The concept of these heterogeneous magnetic circuits opens a very interesting scenario in which in the extreme case, depending on the application, information can be translated in the texture that can accomplish with the lowest effort that particular task. This concept is further sustained by the concept that some classes of magnetic material are able to host multiple textures under the same conditions. In this chapter this concept will be explored taking advantage of the Skyrmion-Domain Wall Pair conversion to put in communication two well established technologies for memory and logic: Skyrmion and Perpendicular Nanomagnetic Logic.

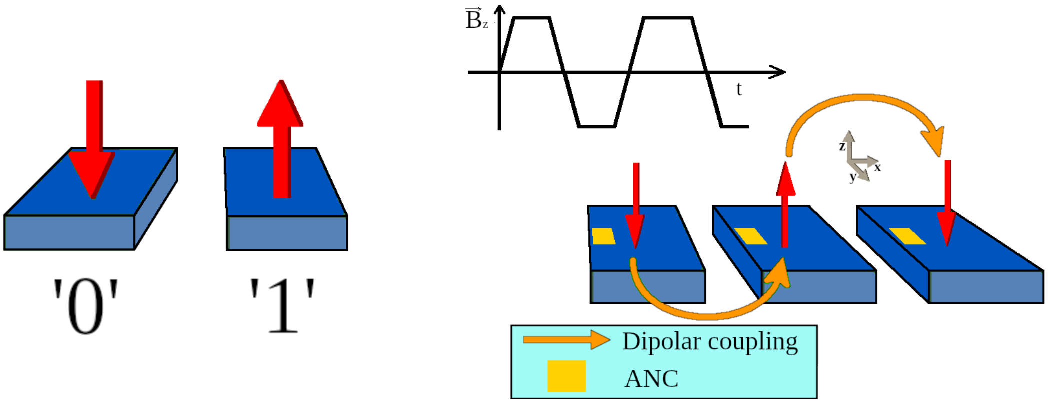
## 8.2 Perpendicular Nanomagnetic Logic

### 8.2.1 Nanomagnetic Logic

Nanomagnetic Logic is a technology that uses magnetic stray fields of nanomagnetic islands to propagate and elaborate information. The idea is an evolution of the Quantum dot Cellular Automata (QCA) technology. Originally, in the proposal of QCA [100], the transmission of information and the elaboration was the result of the coulomb repulsion of electrons belonging to different quantum dots. This technology was very promising for the very high computational speed, the low dissipated power, the absence of interconnections and high-density achievable. Nevertheless the required cryogenic temperatures and difficulties in obtaining uniform devices pushed the research in the direction of substituting the original quantum dots with other objects. A first proposal suggested the substitution of the quantum dots with molecules still using electrical stray fields and coulomb repulsion for information propagation and manupulation (MolecularQCA)[101, 102, 103]. Another proposal suggested to use small magnetic island in the range from hundreds nanometers to few microns to realize logic.[104] The state, now encoded in the magnetization direction of a magnetic island is transferred and elaborated by means of the related magnetic stray field. This technology was at first denominated Magnetic Quantum Dot Cellular Automata (MQCA) [105, 104] but then took the name nanomagnetic logic [106] from nanomagnets, the name given to small magnetic island with uniform magnetization. After the first proposal and theoretical studies, the logic based on majority gate was finally demonstrated at room temperature in 2006 [104]. The material used in the first experimental demonstrations was the Permalloy(Py). This material is a soft-magnetic material with not noticeable anisotropy. The magnetization of the nanomagnets was in plane and the magnetic states were stabilized by means of the island shape. From the first proposal the technology was studied in depth, many proposals were done about clocking mechanisms [107, 108, 109, 110], information stability [111], integration, input and read out techniques [112, 113]. Few years after the first proposal of Nanomagnetic logic based on in-plane magnets, Csaba et al. proposed vertical pillars to obtain an out-of-plane magnetization. The experimental realization of this idea remained lacking until, in 2007, the principle was demonstrated by Becherer et al. with Co/Pt thin films substituting the original nanopillars with plain island with strong perpendicular anisotropy. This technology was named Perpendicular Nanomagnetic Logic.

### 8.2.2 Perpendicular Nanomagnetic Logic

The perpendicular nanomagnetic logic is a technology that takes advantage of the magnetic stray field of magnetic islands with magnetization out of the magnet plane to propagate and elaborate information. [114, 115] The basic element of this technology are nanomagnets with uniaxial perpendicular anisotropy strong enough to allow only two stable magnetic states pointing out of the magnet plane. The nanomagnets are realized starting from thin films with high crystalline anisotropy. The digital information is then encoded in the direction of magnetization.



The propagation of information is based on the principle that taken two nanomagnets the total energy of the system will be lower if the the two magnets are aligned in opposite directions. This coupling is called antiferromagnetic coupling. The properties of the nanomagnets are then locally engineered to create weak spots in order to give a direction to the propagation of information. These weak spots called Artificial Nucleation Center (ANC) are engineered to make the magnet more sensible to stray field generated by neighbor magnets. [114]

#### Clocking mechanism

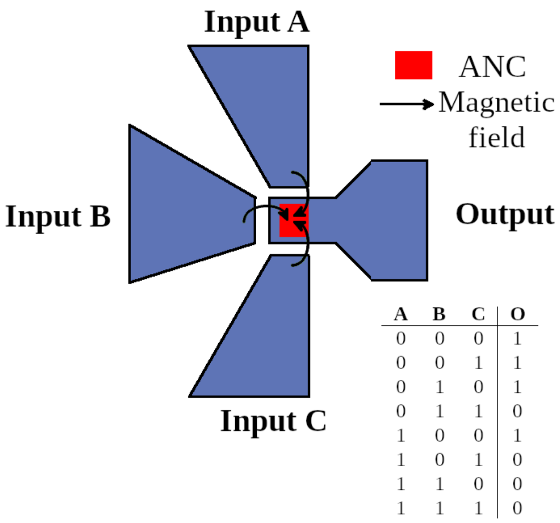
To correctly propagate information an external stimulus is required. This external stimulus is used to trigger the nucleation in the ANC. The common technique to promote the nucleation of information and consequently its propagation and elaboration through the circuit is the application of a varying magnetic field to the complete circuit. This field acts also as a clock signal synchronizing the propagation of information. The external magnetic field is calibrated to be slightly lower, in amplitude, than the switching field of the nanomagnets. Due to this characteristic of the external field, a magnetic island will switch to another state only if a coupling field in the right direction is present. [114] This basic mechanism allows to these circuits to benefit for a global magnetic field that acts as promoter for information propagation [116]. Another possible mechanism for stimulating the nucleation of information is by means of spin hall currents. This technique requires a heavy metal layer beneath the magnet to produce a spin current. The current stimulus exerts a torque on the magnetization of the magnet, pushing it in a metastable state. When the stimulus is removed the magnetization the magnetization will then relax according to the magnetization of the neighbors. This technique differently from the previous allows a local control of magnetization and do not require the generation of a nucleation center to propagate information but a precise synchronization of the current stimuli.[117]

### Logic Operations

The two main operation executed in PNML are the inversion and the majority operation. The inversion operation is executed simply putting two nanomagnets close. The nanomagnet with the ANC closer to the neighbor will be pushed in the state opposite to the other magnet realizing the inversion. The majority gate is realized positioning a nanomagnet with an ANC in the middle of an uneven number of inputs as shown in figure 8.2. When stimulated by the clocking field, the output magnet will switch into a state inverse to the majority of the inputs. The majority logic is a complete logic family and this allows theoretically to the PNML technology to cover all the logic functions realized by the other boolean technologies. From the majority gate is indeed trivial to derive other elementary gates like NAND and NOR fixing one of the inputs. In addition majority logic can be used to realize more complex logic function when more than 3 inputs are used [115, 118]. Both majority and inversion gates were demonstrated experimentally in Co/Pt multilayer films.

The information in Perpendicular Nanomagnetic Logic (PNML) is also propagated by domain propagation. If a magnetic island has n elongated shape the information nucleated at one side of the magnet can be transferred to other zones to be further elaborated. This characteristic is important for PNML and differentiate it from other QCA technologies like Molecular QCA where the transfer of information is realized with an even number of cascaded inversions. The limit of this communication approach is that the expanded domain have to complete its expansion, reaching the other side of the magnet before the switching field changes direction. This time sets together with the nucleation time the critical path of a common PNML circuit [119, 120]

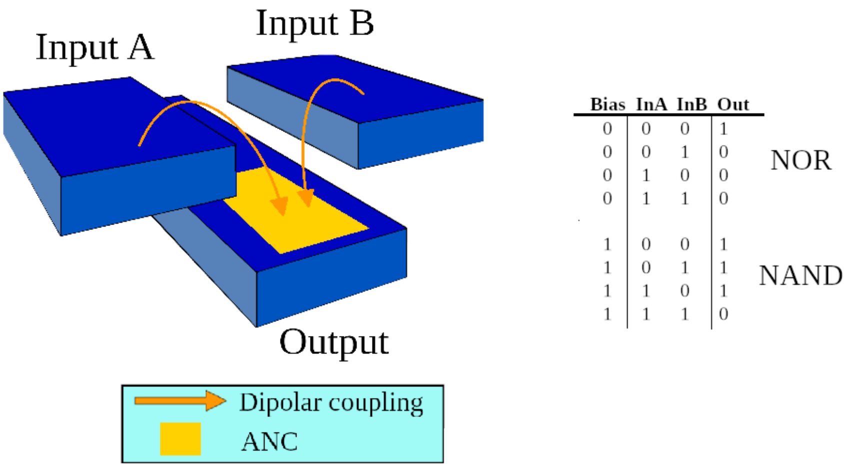
Recently also a 2-input NAND/NOR gate was proposed for PNML [121]. The logic gate is shown in figure 8.3. The inputs placed on a different plane with respect



| A | B | C | O |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Figure 8.2: 3-Inputs PNML majority voter.

to output can influence the nucleation of a new state in the output magnet. Due to the even number of inputs, a bias field is required to produce a result at the output for every input combination. The two inputs are coupled antiferromagnetically to the output. The bias field acts as the third input to favor one of the two stable magnetic states at the output.



| Bias | InA | InB | Out |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Figure 8.3: 2-Inputs PNML logic gate. The function realized by the logic gate is defined by an external bias field in positive or negative z-direction.

### Integration

PNML circuits can be integrated with CMOS technology taking advantage of one of many possibilities of write and read magnetic states. In particular the great advancement of MTJ devices also at industrial level, defined a preferable way to write and read magnetic states. In addition films based on CoFeB/MgO stacks, that proved to be optimal for the realization of MTJ structures , were also recently explored for the realization of PNML gates [121, 122] Moreover the writing and reading operation of magnetic states can also be realized with other devices. The write operation can be realized local with local magnetic fields[123], locally engineered material defects or local current injection[124, 125, 126], taking advantage of electric-magnetic phenomena like SHE. For reading operation other techniques are possible like Hall sensors or anisotropic magneto-resistance sensors. Finally the materials and fabrication processes used in to realize PNML devices are compatible with standard CMOS circuits allowing a good integration with CMOS technology.

## 8.3 Skyrmion-PNML device

Skyrmions with their high stability and small dimensions are more suitable than PNML for memory application. While for PNML, the concept of memory is linked to the magnetization of an entire magnetic island, Skyrmions, encoding the information directly in the magnetic texture, can be packed in smaller dimensions and moved in memory more easily. Their movement is indeed less subjected to pin effects at material defects with respect to domain wall. Moreover, the ease to move a skyrmion on a track, as shown in the previous chapters, suggests their natural application in racetracks [21]. This kind of application allows to reduce drastically the number of device needed for reading and writing purposes with respect to a confined domain with a fixed position, lowering the overall circuit complexity. The number of devices in skyrmion memory devices is then only dictated by the required reading latency due to the serial nature of the read and movement operations. The realization of logic in skyrmion on the other hand is quite difficult and requires a great control of the skyrmions inside the track. Moreover, the distribution of information inside the circuit, as shown in the previous chapters, requires a particular care. Multiple duplication gates are needed and their employment requires a not negligible consumption of energy with respect to the energetic costs of movement. Finally, the movement of skyrmions is limited to a single plane. There are not solutions in literature designed to allow to skyrmions to move in multiple layers. This factor limits at the routing phase of the circuit the design. Often multiple crossing and additional controls are required to allow to some information to move through the circuit, as shown in section 6.2.3. On the contrary, PNML, has great advantages in the realization of logic functions. The possibility to use multilayers for circuits [127] gives to the design many degrees of freedom for the realization of complex circuits [128]. Furthermore, the possibility to guide the entire circuit with a single stimulus, i.e. the global varying magnetic field explained in section 8.2.2, allows to the circuit to reach a great efficiency in logic computations, especially for complex circuits where a single magnetic field can guide thousands of gates. [116] Finally the distribution of information in PNML is easier with respect to skyrmions. The possibility of routing magnetic signals on different layers gives to the design different degrees of freedom to easily distribute information along the circuit. [127] For this reason the goal of this chapter is to find a feasible way to connect these two technologies. This in order to enable the design of a device in which the memory is demanded completely to skyrmion while the logic is left only to PNML allowing complex elaboration of information stored at minimum effort. The structure of the complete device is shown in figure 8.4.

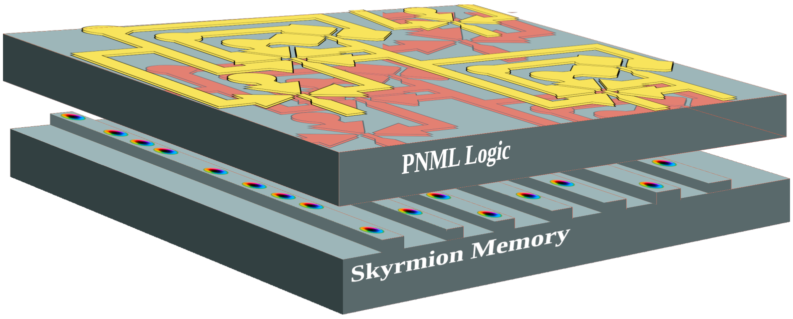


Figure 8.4: Skyrmion-PNML device.

As shown the device is thought as a multilayer structure in which a first layer is specific for memory. Here the information is stored in form of skyrmions. The kind of memory employed is a skyrmion racetrack memory. This layer is designed as the storage of the device. On top of this first layer, the logic layers in which information coming from memory can be elaborated and processed.

### 8.3.1 Information stability

The study made on skyrmion-PNML interface started from the study of information stability on skyrmion memories under varying magnetic fields. This parameter is very important as it will shape the final device determining if the presence of a varying external field can affect the information stored. It is unlikely that all information stored in memory are processed at the same time. For this reason, the information stored in form of skyrmions in memory during the PNML operation will be affected by the external magnetic field. In [129] Buttner et al. described analytically the skyrmion energy model, allowing to predict the stability of skyrmions in different materials parameters configurations. As shown in the article, in different classes of material the skyrmion state is stable for wide ranges of external applied fields, higher than 100 mT in magnitude. The material chosen for the study is the W/CoFeB/MgO described in [52]. The parameter of the material used are reported in table 8.1.

Table 8.1: Parameters used for micromagnetic simulations and current distribution computation

| SIMULATION PARAMETERS | | |
|---|---|---|
| Saturation Magnetization [52] | 1e6 | $\mathrm{A\,m^{-1}}$ |
| Uniaxial Anisotropy Constant [52] | 8e5 | $\mathrm{J\,m^{-2}}$ |
| Exchange Stiffness [52] | 2e-11 | |
| Damping constant [75] | 0.03 | |
| Spin Hall Angle [63] | 0.4 | |
| Film resistivity [63] | 165 | μΩ cm |

The material is capable of hosting stable skyrmions [9] and it is also compatible with PNML [122].

The first study conducted to verify the effect of external magnetic fields on information, is the verification of the stability of the skyrmion in a wide plane. The structure hosting the skyrmion was a plane with dimensions $256\,\mathrm{nm}^2 \times 256\,\mathrm{nm}^2$ and thickness 1 nm. The isolated skyrmion was imposed as starting magnetization and then relaxed to reach a stable magnetization state. The isolated skyrmion had topological charge equal to 1 and core oriented in the negative z direction. Starting from this state an external field of increasing intensity in the negative Z direction was applied. Figure 8.5 shows how the skyrmion modified its shape in response to the external field.

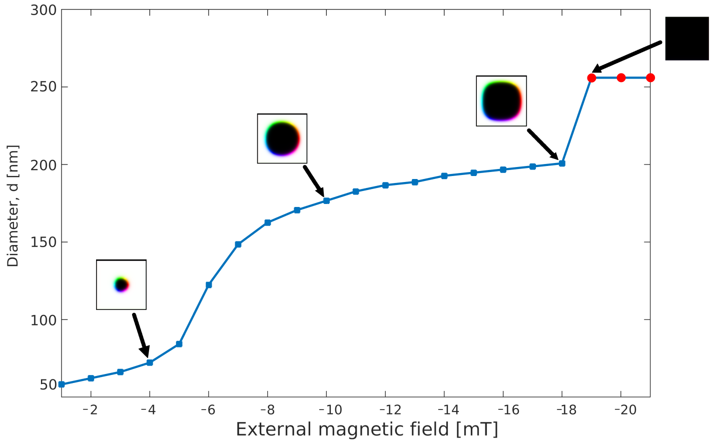
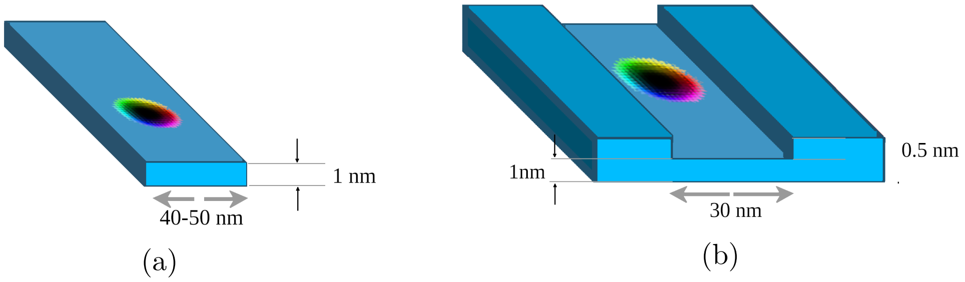


Figure 8.5: Skyrmion radius evolution with an applied external magnetic field. The skyrmion is confined in a plane with dimensions 256 nm × 256 nm

As can be seen from the figure 8.5, the skyrmion has the tendency to expand due to the influence of the external applied field. The external applied field promotes the expansion until the repulsion from the borders limits the maximum radius. This force is strong enough to avoid the complete loss of the texture until a field of −18 mT is reached. After this field value the repulsion from the borders is not enough to keep the texture confinement and the sample magnetization is completely switched in the negative direction. Regarding the positive field the skyrmion can withstand fields up to 33 mT after which the texture is lost. These values was measured imposing an external positive field for 2 ns. It is important to notice that the radius is increasing at every external field.



(a) (b)

Figure 8.6: Skyrmion confinement structures. **(a)** plain racetrack **(b)** curbed racetrack

After the simulations in a square plane, the configuration was tested with varying field in a confined geometry. The first structure tested was a simple track with varying dimensions. The dimension tested were 40 nm and 50 nm. These dimensions were chosen as these are the smallest width admitting stable skyrmion states. For a plain track with dimensions lower than 40 nm the skyrmion state is not stable anymore in the track. In these simulations, the structures were tested with periodic boundaries condition along x equal to 1. The field was varied from −40 mT to 40 mT with a step of 1 mT. The simulations, after the skyrmions were relaxed in a stable states, were run for 2 ns. The results for the negative fields are shown in figure 8.7. It's interesting to notice that a for a track with dimension 40 nm the skyrmion cannot withstand any negative value of external field. The texture in this case expands immediately and touches the borders losing its topological charge. The 50 nm case is more stable, in this case for negative fields as shown in figure the configuration is stable up to 17 mT, after, the repulsion from the borders is not enough to contain the skyrmion texture and the skyrmion is lost. The resistance to the positive fields was higher compared to the previous cases, the texture was indeed stable for all the positive external field tested, i.e. up to 40 mT. Finally, a geometry like the one used in the LIM exploration was tested in the same way. The

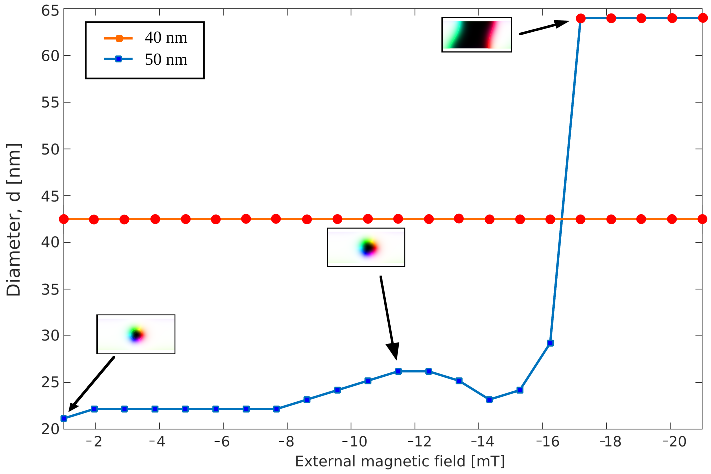


Figure 8.7: Skyrmion radius evolution with a negative external field. The skyrmion is confined in a 40 nm strip and in 50 nm

geometry is shown in figure 8.6b. The curb of the tested geometry is wide 30 nm and deep 0.5 nm . This geometry is the reference used in the previous studies as basic track for memory. As shown in the figure 8.8, the skyrmion confined in the curb, subjected to an external field has the tendency to maintain its shape for even higher fields in the tested range both for negative and positive fields. Thanks to the curb confinement, the skyrmion is able to withstand a wider range of negative external magnetic fields.

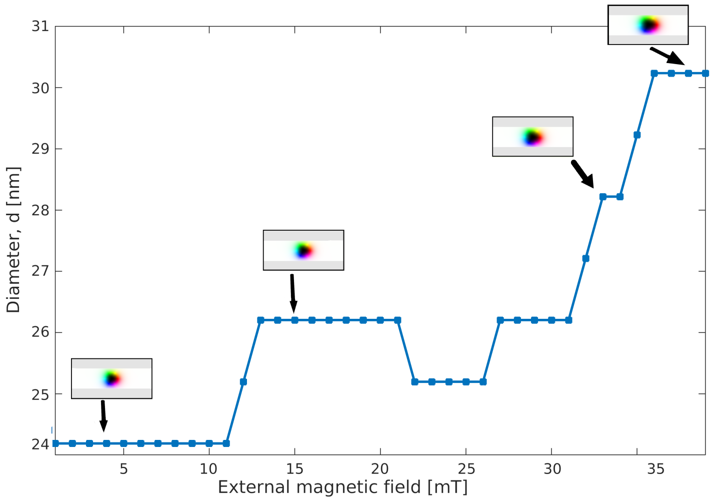


Figure 8.8: Skyrmion radius evolution with an applied external magnetic field. The skyrmion is confined in a curbed track wide 30 nm and deep 0.5 nm

The field range that was found in this study is important in defining the working conditions for PNML.

### 8.3.2 Skyrmion to PNML conversion

The next step in this study is the definition of the interface between Skyrmion and PNML. The structure to execute the conversion is the one showed in figure 8.9.

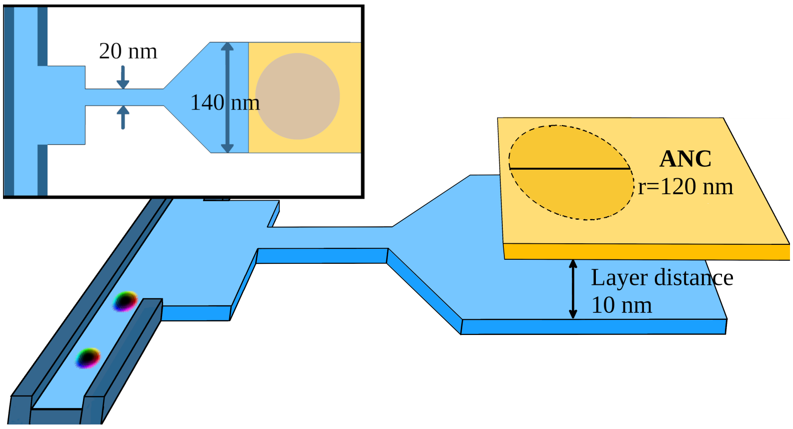


Figure 8.9: Skyrmion-PNML conversion structure. The yellow structure on top is the input for the PNML logic

The initial stage of the conversion is represented by a conversion that takes advantage of the conversion phenomena showed in [33] by Zhou et al. The skyrmion is pushed by means of a current in the direction of a constriction with a width lower than the radius of the actual skyrmion. When the skyrmion collides with the constriction the topological protection is lost and the skyrmion is converted in a DW pair. The information is now encoded in a domain. In this state the domain is used as a seed to switch completely the remaining part of the gate. The domain is expanded by means of an external magnetic field in the direction of the core polarization of the skyrmion. Then, the stray field linked with the input information is strong enough to be used to switch a nanomagnet on another layer placed in correspondence of the larger part of the structure. The top nanomagnet in the conversion structure shown in figure 8.10 is switched accordingly to the magnetization of the bottom structure to generate a valid input for the PNML circuit. The top nanomagnet is designed with a nucleation center. The same field used for the expansion is used to nucleate in the top nanomagnet, the new information. The first external field is followed then by a second wave with the opposite polarity to allow also to the positive magnetization direction to propagate in case the original information was not a skyrmion but a hole. The information, now transferred to a nanomagnet can be completely processed starting from the top layer in PNML.

### 8.3.3 PNML to Skyrmion conversion

The inverse conversion from PNML to skyrmion is executed with the structure shown in figure 8.10.

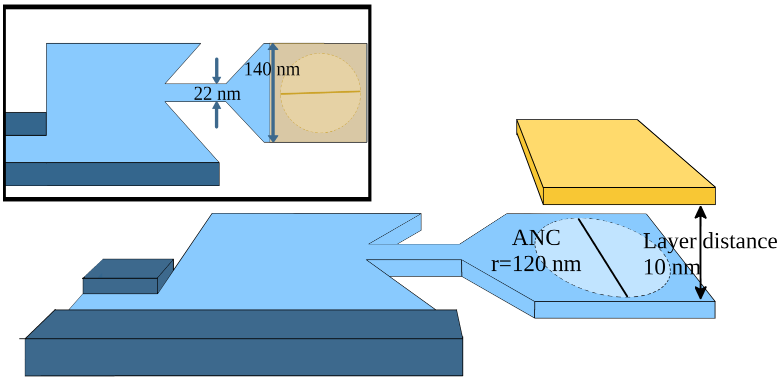


Figure 8.10: PNML-Skyrmion conversion structure. The yellow structure on top is the the PNML logic

The information after the PNML elaboration is expected to be ready in the top nanomagnet. As shown the top nanomagnet is placed on top of another structure, designed to translate the information in skyrmion form for storage purposes.

In the presented structure the magnetic coupling between the top and the bottom islands is strong and the magnetization value can be successfully transferred. To help this process, the structure on the bottom presents an ANC in correspondence of the top nanomagnet island. The ANC has been calibrated to allow to the global clocking field used for the common operation of the PNML to nucleate the top information from the ANC in the bottom island. The same field will then expand the domain to the border of the constriction. The constriction stops the expansion of the domain guided by the field. To generate a new information in form of skyrmion now a current pulse is used. The current pulse allows to push the domain out of the constriction where a new skyrmion will be stabilized. In case the information at the output of the PNML is positive, this last passage does not produce any domain and a hole is left in the memory, encoding a logic 0.

### 8.3.4 Methods

The structure has been verified with micromagnetic simulations using Mumax3, finite difference simulation software. The basic stack used for simulations is W/-CoFeB/MgO and the parameters for simulations are reported in table 8.2. The reference structure has a thickness of 1 nm.

Table 8.2: Parameters used for micromagnetic simulations and current distribution computation

| SIMULATION PARAMETERS | | |
|---|---|---|
| Saturation Magnetization [52] | $1 \times 10^6$ | $\mathrm{A\,m^{-1}}$ |
| Uniaxial Anisotropy Constant [52] | $8 \times 10^5$ | $\mathrm{J\,m^{-2}}$ |
| Exchange Stiffness [52] | $2 \times 10^{-11}$ | |
| Damping constant [75] | 0.03 | |
| Spin Hall Angle [63] | 0.4 | |
| Film resistivity [63] | 165 | µΩ cm |
| ANC Anisotropy | $5.6 \times 10^5$ | $\mathrm{J\,m^{-2}}$ |

### 8.3.5 Micromagnetic simulations

The simulation of the conversion process is shown in figure 8.11. The simulation shows only the cases in which the information at the input is a logic 1, i.e. a skyrmion is present. The logic zero case, equivalent to the absence of a skyrmion are trivial. No conversion is needed for the operation.

The conversion of information starts with a current in the direction of the constriction. The current intensity as shown in figure has a magnitude of $3.5 \times 10^{10}\,\mathrm{A\,m^{-2}}$. The current used in this phase can be even lower as long as the current density remains higher than the depinning current of the skyrmion. After a time equal to 0.5 ns the skyrmion is in proximity of the constriction. A current pulse is now applied to the structure to overcome the repulsion from the borders and overcome its topological protection. The skyrmion is then translated in the constriction in a domain enclosed by a DW pair.

Now, a negative field of amplitude 24 mT is applied to the structure for at least 3 ns. The field produces an expansion of the domain in the complete structure. While the domain is expanded on the right , on the left side of the constriction the domain remains pinned. This condition is important to avoid the expansion of the domain inside the memory track. After the section on the right side of the constriction is completely nucleated with a polarity equal to the core of original skyrmion, negative in the simulations, also the top magnet starts to nucleate a new domain with a negative polarity. The same nucleation field, used previously, promotes the nucleation of a new domain starting from the ANC Finally the field completes the nucleation of the domain in the area after 3 ns. The negative pulse is then followed by a positive pulse of the same amplitude. This second wave allows also the nucleation of the positive information propagation to the top domain in case the input data for the block was a 0, encoded in the present device with the absence of a skyrmion at the input.

After the simulation of the Skyrmion to PNML conversion the opposite transition has been tested.

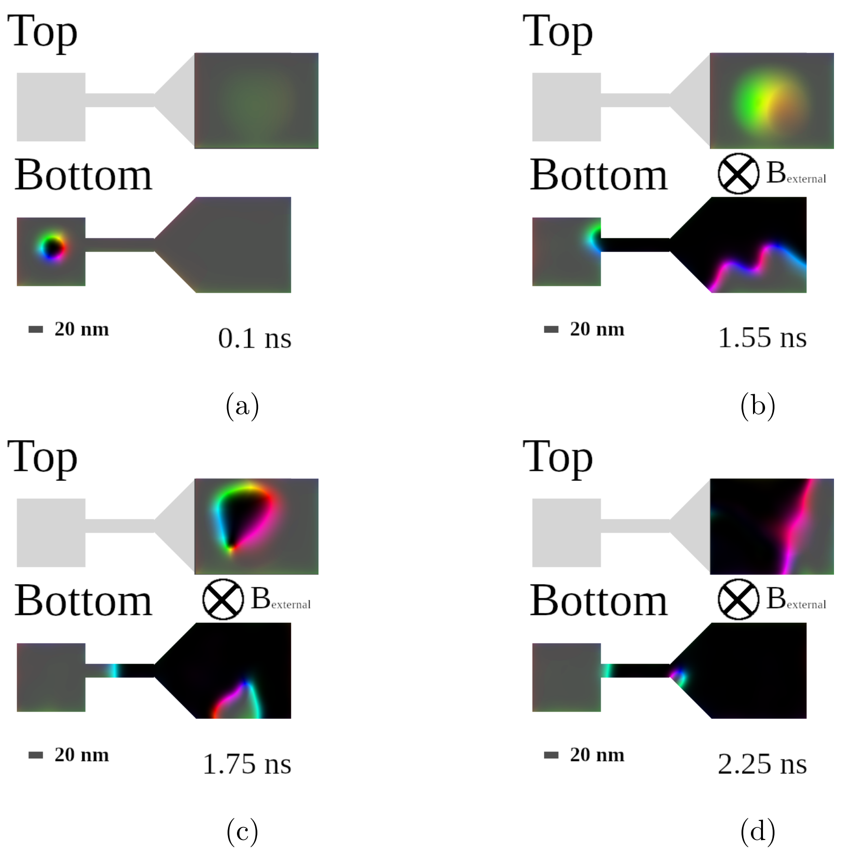
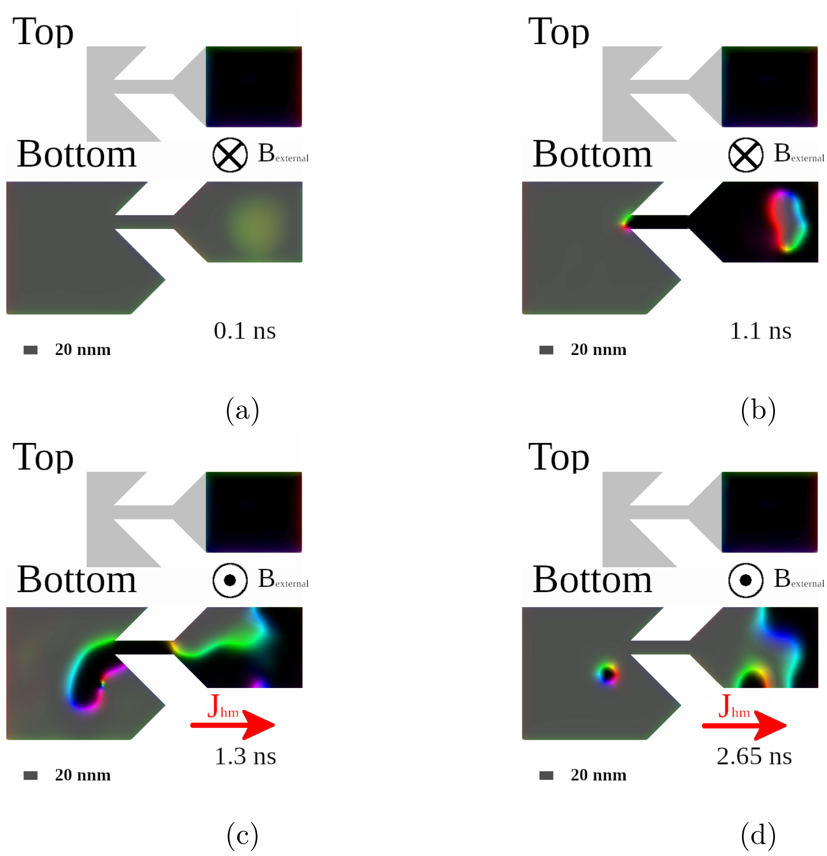


Figure 8.11: Skyrmion to PNML input conversion. The skyrmion in proximity the constriction (a) is pushed by means of a current and is converted in a domain(b). The domain is the expanded by means of a magnetic field. A nucleation is triggered in the top structure. A domain in the top structure is nucleated (c). The domain expands in the top structure (d).

In figure 8.12, the simulation of the reversed conversion is shown. In this second case the input is the magnetization polarization of the top magnet. This magnetization in the circuit topology shown in figure 8.4 is the result of the PNML circuit. The injected information can be both a copy of the original information or an elaborated version of it. The gate will generate at the end of the conversion a new skyrmion in the track depending on the polarity of the magnetization in the top nanomagnet. As before only the negative case will be shown. The positive case that should not produce any skyrmion at the output is trivial. The starting condition in the simulation is the top magnet completely magnetized with negative polarization and the bottom completely magnetized in the opposite direction. A negative magnetic field of intensity 24 mT is then applied to structure. On the bottom the gate provided with an ANC a domain with the same polarity as the top input starts nucleating. In a similar way as the previous case the new domain nucleated



(a) (b) (c) (d)

Figure 8.12: PNML to skyrmion conversion conversion. The PNML output is in the bottom direction and influence the ANC in the bottom structure(a). A domain is nucleated in the bottom structure with the same polarity as the top input (b). The domain expands is the output structure by means of a current. A positive field is applied in this phase (c). A new skyrmion domain detach from the constriction. The skyrmion is pushed away from an external current (d)

expands reaching the constriction where it remains pinned at the discontinuity. At this point a new skyrmion should be produced by the operation. The nucleation of a new skyrmion requires a local stimulus strong enough to push the domain out of the constriction as explained in the previous section. Therefore with the pinned domain wall at the constriction, a current density of $50 \times 10^{10}\,\mathrm{A\,m^{-2}}$ is applied at the structure for 275 ps. The domain in response to the stimulus elongates out of the constriction. To favor the generation of a skyrmion from the elongated domain a positive field is applied to the structure. Consequently, the elongated part of the domain detaches and a skyrmion is generated. The newly nucleated domain now oscillates in the constriction. To reduce the oscillation and push the skyrmion in the direction of the memory track, a current of $5 \times 10^{10}$ is applied. The information can now be memorized in the racetrack memory. The time of the conversion with the described stimulus is of 4.275 ns. The conversion time comprehend also a relax time, in absence of current between the elongation 8.12c and the skyrmion stabilization and movement, figure 8.12d, of 2 ns.

### 8.3.6 Energetic evaluations

To evaluate the power and energy cost of the conversion, the power required by the current pulse and by the magnetic fields required by the conversion were considered.

The results are shown in the following table:

Table 8.3: Energy performance of Skyrmion-PNML conversion

| | **Power** | **Energy** |
|---|---|---|
| Skyrmion - PNML | 15.2 µW | 4.56 fJ |
| PNML-Skyrmion | 95.04 µW | 26.3 fJ |

While the calculation of the power linked to the current pulses can be obtained considering directly the dimensions of the gate and the resistivity of the material, the evaluation of the power and energy linked to the external magnetic field generation requires additional considerations on the circuit in which the conversion gate is implemented. The fields applied are not local to the gate but are applied globally to the complete circuit. Therefore, in the evaluation of the power linked to the external magnetic field the density of the gates is considered as well as the total area occupied by the circuit. Not knowing at this stage, the size of the memory and the number of parallel input ports from the memory to the PNML circuit the evaluation is complex. Fixing the area of the circuit, indeed, the power required by the single operations scales with the number of gates the magnetic field is able to guide at the same time. For this reason, the case in which the external magnetic field guides a circuit with area 1 $\mathrm{mm}^2$ is taken as reference for the computation. Then, 1024 conversion gates are considered. These two values, considering a parallel architecture represents a worst case scenario in which the area occupied by the logic circuit is huge and the number of parallel inputs to the logic is limited. As an example AES cipher presented in previous chapter process 128 bit per LIM unit, in a real circuit an high number of these units working in parallel is desirable.

In the considered scenario, the power required to guide the circuit is equivalent to 3.3 mW. The reference power density was computed starting from the value given by Becherer et al. in [116]. Now, to derive the power required per conversion, this value is divided by the number of devices, 1024 in our hypothesis. The value obtained in the presented situation is 32 µW per bit read. Finally it should be considered that the translation in PNML and back can happen during the normal operation of the circuit. For this reason the energy cost of this conversion should also be shared during operation with the processing unit on top.

Finally regarding this worst case also the energy linked to this conversion was computed. The energy linked to this transformation is 160 fJ per bit. The value was obtained multiplying the power obtained above with the time required by the gate to nucleate a new information considering both the negative and the positive waves connected with the transformation operation with length 3 ns each.

Another nucleation mechanism that can be employed in case a small number of input devices is required, is the method proposed by Bhowmik et al. in [117]. The authors propose to use spin currents to push some nanomagnets in a meta stable state and then let the system evolve according to the input. This method differently from the global magnetic field is affecting only the magnets that are interested by the current flow. For this reason the efficiency is expected to be high for a small number of devices and start to decrease linearly with the increase of this quantity. The authors in the article reports a working experimental value of $3 \times 10^{11}\,\mathrm{A\,m^{-2}}$ for a successful switching of magnets. To estimate the power consumption of the presented structure we can consider the nucleation of an input magnet of width 140 nm and length 256 nm like the one used in simulation of the Skyrmion to PNML conversion showed in figure 8.10. The resulting power for this conversion is 26 µW per bit converted. This value is lower compared to the solution taking advantage of the field but does not scale in the same way with the increase of the parallelism of the circuit as the solution relying on external magnetic fields. Comparing the two solutions together, the solution that use currents for nucleation is therefore convenient up to a number of devices equal to 1240. After this number, the most convenient solution, regarding the power consumption of the circuit, is the use of external magnetic field. Finally, it is important to notice that this value is valid for the case presented but that should be recomputed depending on the area of the circuit and consequently the power required for the magnetic field generation.

## 8.4 Final considerations and future works

In this chapter, we demonstrated an interface between PNML and skyrmion. This interface can provide to PNML an high density storage for computation and a preferable way to store computed information. This chapter shown that the interface has good conversion timings and energy, providing a preferable way in which PNML can receive inputs and provide outputs. The demonstration of this connection opens the way to a new kind of circuits in which different magnetic technologies are mixed and can be optimized for single applications without the need of finding a trade-off between the different tasks realized with a technology. In particular looking at the skyrmion case, a strong effort has been done in the direction of obtaining a skyrmion movement with almost no deviation from the current direction, while for most of the logic with skyrmions proposed in literature, this deviation is a requirement. In fact, the removal of SkHE is going to inhibit almost completely the possibility to realize logic, allowing at the same time to reach a tremendous access time for the memory. Allowing a direct connection to another magnetic technology the optimization for skyrmion technology can focus only on the optimization for memory applications, leaving the elaboration of information to PNML technology. In the same way, this technology used for elaboration can be optimized for logic applications, not requiring anymore an high information stability provided in these hybrid circuits by skyrmions. Finally, another important characteristic of the proposed structure is the compatibility of the two technologies in terms of external stimuli and materials employed that can be an advantage from the point of view of manufacturing process. The realization using a single material, can indeed give an important help in the integration process of these circuits. In the future a deeper study on the possible applications of these hybrid circuits should be conducted. Circuits for specific applications should be designed and evaluated to evaluate the benefits the presented proposal can give compared to other technologies. The reference scenario for the application of this circuit is the computing-near-memory approach were the information is elaborated very close to the storage. The implementation of real algorithms will allow the estimation in a real scenario of the cost of the proposed interface. Finally, it will be important to understand how a circuit with the presented structure can be beneficial to a complete elaboration system.

# Chapter 9

# Final conclusions and prospects

In this thesis the problem of computation in memory with skyrmions was addressed. Starting from a new design methodology for logic-in-memory circuits, different implementations were explored and evaluated. The results from the performance evaluations suggest that skyrmion technology in the conditions explored still needs some study and improvement for what concerns the energetic performance. At the same time, the skyrmion technology proved to be excellent in the density achievable from the proposed circuits. In all the proposed solutions, but especially in the CAM and AES case, the implementations proved to be able to pack a great number of gates in a small area. This aspect is particularly interesting in logic-in-memory implementations where the impact on the die area is usually high. In the skyrmion case this achievement is even more important. In this scenario, the additional area occupied can have two negative effects: diminish the density of information, that is one of the most important characteristics of skyrmion memories and slow down the read and write process of the memory. This process is based on shift operations and a lower density can represent a lower read and write speed making the memory more difficult to be integrated in real systems. For the future it is important to study further the problem of logic-in-memory with skyrmion. The characteristics underlined in this thesis show a good potential for the current technology especially in systems in which the latency is not a critical parameter. In this direction, it will be important to study both the possible optimizations applicable to the proposed systems and the material parameters that can further optimize their characteristics. Finally, regarding this first part, it is important to underline that the destiny of such logic application is strictly linked to the success of the skyrmion memory. The device by themselves does not give enough advantages to be proposed as stand-alone logic devices. Similarly to what happened to bubble memory, the main core of skyrmions remains still the memory and without the possibility to employ skyrmion primarily on memory, at the actual state of the art, the proposals of this thesis do not have a real reason to be used. This suggests that the research in the direction of logic-in-memory applications should focus on adapting itself to the best possible implementations for memory for which the proposed architectures should represent an expansion. In this direction the second proposal of this thesis was done. The conjunction of PNML and skyrmions in the same device, allows to connect to a good technology for memory with a good technology for distributed logic. The generation of domains for skyrmion is a very natural process in magnetic materials usually employed for skyrmion devices. As shown in literature in [34], also the conversion from plain domains to skyrmions is possible. This, linked to the recent advance in PNML technology [122, 121, 128], creates a very exciting scenario in which new kind of circuits that depend less on electric control are possible and can completely take advantage of the best characteristics of magnetic circuits. The last proposal of this thesis is a starting point in this direction and shows a simulative proof of the integration between PNML and skyrmion technology. It will be important in future to test this interface against real systems designed for specific problems and understand if it is really worth it compared to other solutions. In addition to that it will be also interesting to understand if this same approach of communication between skyrmion and other magnetic technologies is usable for other computing paradigm that in the current moment are attracting great interest from a wide part of the electronic community. Finally one last point, that is important for the development of the proposed approach, is the experimental demonstration of the elements that in this thesis were used as building blocks for all the architectures. In fact, in addition to skyrmions moving in a track, also the demonstration of logic devices based on this technology will be important for the development of logic-in-memory applications. This is the most important benchmark to understand if a technology has the potential to be used, if it needs further research or if it is not worth to be used in computation environments.

# Acronyms

| | |
|---|---|
| **LIM** | Logic-In-Memory |
| **LLG** | Landau-Lifshitz-Gilbert |
| **DMI** | Dzyaloshinskii–Moriya interaction |
| **MOKE** | Magneto-Optical Kerr Effect |
| **XMCD** | X-ray magnetic circular dichroism |
| **DW** | Domain wall |
| **VCMA** | Voltage Controlled Magnetic Anistropy |
| **SOC** | Spin Orbit Coupling |
| **STT** | Spint Transfer Torque |
| **SHE** | Spin Hall Effect |
| **SkHE** | Skyrmion Hall Effect |
| **LSB** | Least Significant Bit |
| **MSB** | Most Significant Bit |
| **SL** | Search Line |
| **RAM** | Random Access Memory |
| **CAM** | Content Addressable Memory |
| **LIM** | Logic In Memory |
| **IOT** | Internet-of-Things |
| **AES** | Advanced Encryption Standard |
| **TDES** | Triple Data Encryption Algorithm |
| **CMOL** | CMOS-nanowire-molecular |
| **MTJ** | Magneto Tunnel Junction |
| **PNML** | Perpendicular Nanomagnetic Logic |
| **ANC** | Artificial Nucleation Center |

| | |
|---|---|
| **QCA** | Quantum dot Cellular Automata |
| **MQCA** | Magnetic Quantum Dot Cellular Automata |
| **SkHE** | Skyrmion Hall Effect |

# A quick look at the past: Bubble memory and logic

Magnetic bubbles were in the 1970s and first half of the 80s a popular technology for memories. This technology, based on micrometer sized magnetic bubbles, attracted great interest and reached also the commercialization in memory chips of dimensions also bigger than 1Mbit [130]. As it will be shown later, this technology had some important basic elements in common with skyrmion technology. This particular memory technology was abandoned commercially not much time after its adoption for the great advancement that other competitor memory technologies had [131, 132]. Nevertheless, this technology attracted a lot of research and effort also in the direction of logic devices. In the following small appendix a small overview of the technology will be presented. Some logic-in-memory implementations involving bubble technology are shown. Finally a comparison between skyrmion and magnetic bubble technology is presented in order to show how some of the ideas are still valid in the presented context of skyrmions.

## Magnetic bubbles

In 1960, Koy and Enz demonstrated experimentally the realization of magnetic bubbles [133]. These magnetic textures can be defined as round shaped domains in magnetized media with perpendicular anisotropy. Usually stabilized with external magnetic field, these objects have dimensions between 1 and 100 µm. It is important to notice that this particular magnetic textures are stabilized for the most majority by dipolar interactions, differently from skyrmions that are found in system with strong DMI. These textures attracted great interest when in 1967, Bobeck et al. proposed to use the bubbles as means to store information in memories [134]. This initial idea stimulated a great interest in research, leading also to the realization of commercial memories based on this technology [130]. In the 1970s, magnetic hard drives were already a solution employed in a lot of systems as main memories. Differently from that technology, based on information stored in static domains on a mechanical rotating disk, in the bubble memory the information encoded in small magnetic bubbles is moved along the circuit by means of magnetic fields.

The information is encoded in the presence, "logic 1", or absence, "logic 0", of the magnetic bubble. The information is then moved along the circuit following a basic concept of magnetic attraction and repulsion: a local magnetic field pointing in a direction opposite to the core can be used to attract the bubble. In the same way a magnetic field with the same direction as the core can be used to repel it. The principle is shown in figure 9.1.

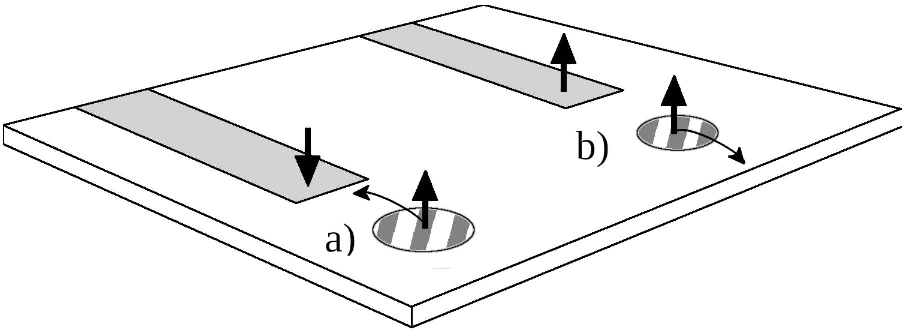


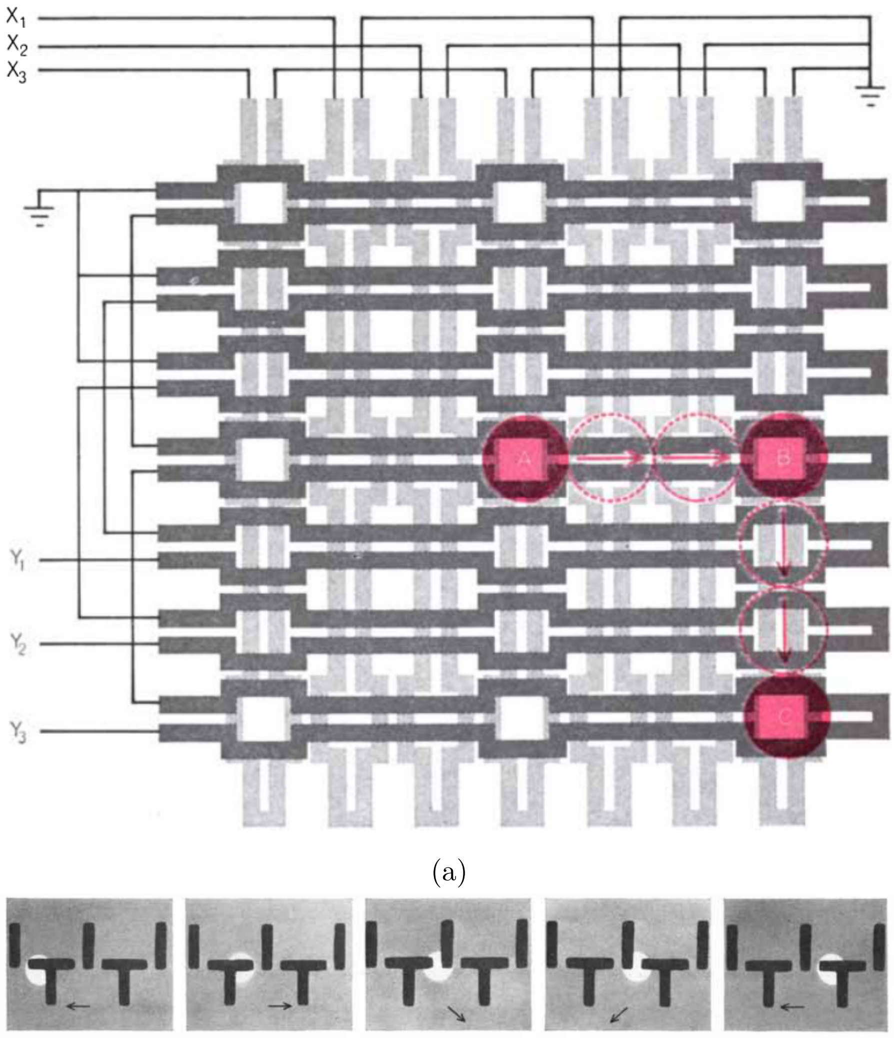
Figure 9.1: Basic movement mechanism of magnetic bubbles. a) When the magnetic field at the end of the bar points in the opposite direction of the bubble core the object is attracted. b) When the magnetic field at the end of the bar points in the same direction of the bubble core the object is repelled.

Starting from the basic principle shown above, the principal techniques to move a bubble were 2:

- Use shaped magnets placed on top of the magnet plane to produce local stray fields able to repel and attract bubbles. The magnetization of these elements was rotated with a global rotating field. Figure 9.2a.
- Generation of local magnetic field with shaped wired to induce the movement of the bubble. Figure 9.2b.

These techniques mixed allows to move bubbles along a circuit and precisely control their movement. Nucleation and annihilation of the element are usually obtained by means of single coils. If a current strong enough is pushed through the coil the magnetic field generated can generate a new bubble or completely annihilate one that is at the coil center.

To realize magnetic memories one of the most used technique to move the bubbles was the global rotating field shown above that imposes to the circuitry a natural synchronization. To keep information stored the bubbles representing part of the data were trapped in minor loops in order to not lose information during the normal circuit operation.[136].



(a)

(b)

Figure 9.2: Moving mechanism for bubble logic. Images reproduced from [135]

# Appendix : Magnetic bubbles logic

With the development of more advanced devices for the movement, also the possibility to realize logic was investigated in research [137, 138, 136, 139]. In particular, the natural repulsion of magnetic bubbles was employed to obtain logic operations between different bits of information traveling through the circuit. In figure 9.3 the most common operations are presented

As shown, except for the majority gate, the basic phenomenon used for computation is the repulsion between elements. The AND-OR, joined with the deflectors that act as the NOT gates, realize complete the logic family. In addition to the presented gates also the mechanism for generation and annihilation of bubbles is

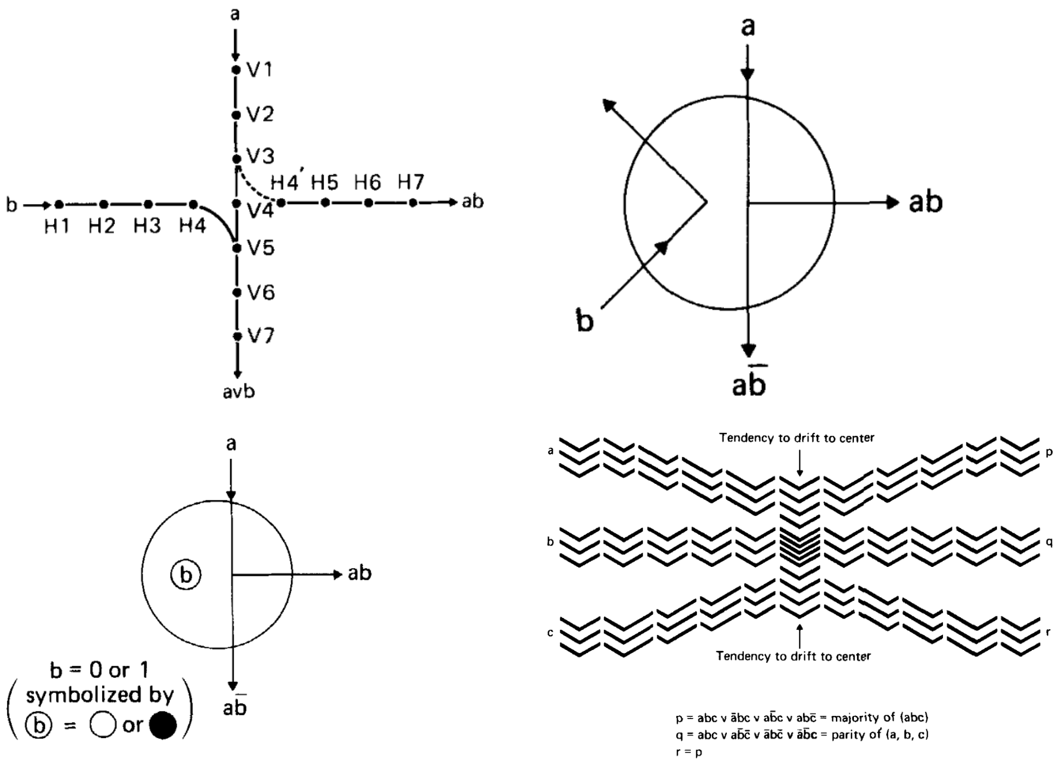

a
V1
V2
V3
H4'
H5
H6
H7
b
H1
H2
H3
H4
V4
ab
V5
V6
V7
avb
a
ab
b
$a\bar{b}$
a
b
ab
$a\bar{b}$
b = 0 or 1
symbolized by
b = ○ or ●
Tendency to drift to center
a
p
b
q
c
r
Tendency to drift to center
$p = abc \vee \bar{a}bc \vee a\bar{b}c \vee ab\bar{c}$ = majority of (abc)
$q = abc \vee a\bar{b}\bar{c} \vee \bar{a}b\bar{c} \vee \bar{a}\bar{b}c$ = parity of (a, b, c)
r = p

Figure 9.3: Magnetic bubble logic gates. a) The AND/OR gate produces at the bottom side the OR results and on the right the AND result between the two inputs comping from the top and left track. The operation is executed with the natural repulsion of bubbles. b) and c) Dynamic and static deflectors. Both the elements determine the path of the input bubble present on input a based on the bubble present in b. The information stored in b can be dynamically or statically placed at the b input. d) Majority gate based on chevrons stacks. Images reproduced from [136]

required to realize the logic.

## Logic-in-memory

Similarly to what discussed about skyrmions, also for bubble memories the employment as a standalone logic device was not convenient with respect to electronic computational devices. This because the maximum frequency of movement of bubbles was limited to 100 MHz and for the fact that produce big fan-outs with bubbles is a quite challenging and expensive task. For this reason all the proposed logic with some exception [138, 139] not discussed in this little appendix, were thought as part of logic memories. The principle was that a bubble memory could communicate directly with these logic gates without the need of any electrical conversion

[136]. Two of these proposals will be discussed in the following.

## Reconfigurable arithmetic gate

In [137], the authors showed the realization of a reconfigurable operation by means of generation and selection of minterms. In the proposed logic the first part of the gate generates the minterms starting from the basic logic, and the second that can be reconfigured with the help of static deflector sums up the statically selected minterms.

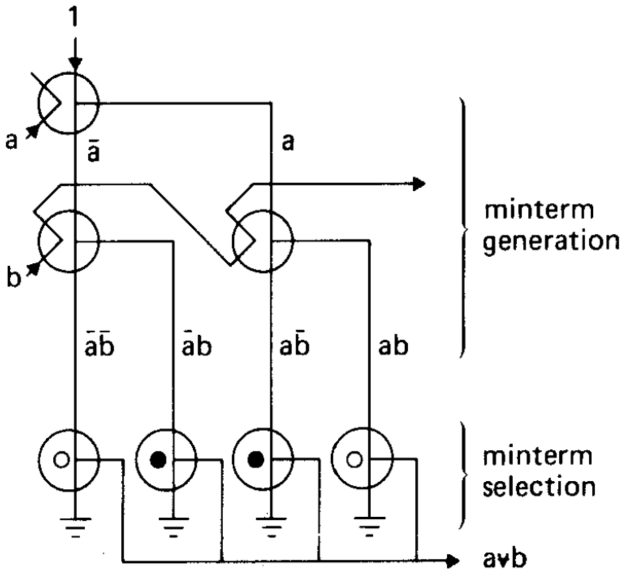


Figure 9.4: Bubble logic based on minterms. Image reproduced from [137]

A very interesting aspect of this logic is the fact that A and B, the inputs of the logic, influence the behavior of the gate without being modified by the circuit itself. This concept is trivial but fundamental for the realization of effective logic in memory implementations: if the information is lost in the processing the unit loses its memory functionality.

## Associative search

Another very interesting yet simple approach to logic-in-memory with bubbles was proposed in [140] where a proposal on associative search was done by Lee and Chang.

This kind of search operation is very interesting and recalls the concept applied for skyrmion in chapter 3. The information traveling through the R input is at first duplicated by means of a dynamical deflector. The information present also in its

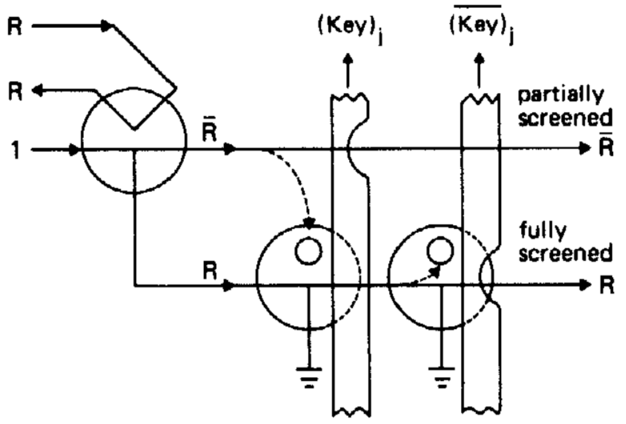


Figure 9.5: Bubble associative search. Image reproduced from [140]

negated version is not guided under two wires that encode the key in a magnetic field. In the presented schema if the key bit is equal to the bit encoded in R the bubble can travel further. If the values are not equal the R bubble is used to block one of the two deflectors in a position that will deviate the next input to an annihilation point. The basic concept here is that a single wrong key-input pair will block all the other operations until a reset of the gate is performed.

## Skyrmion and magnetic bubble technology

Skyrmions technology relates strongly to magnetic bubbles and in a way the latter can be thought as a very close relative to the former. Two main similarities enforce this relation: the local form of the information and the repulsion that bubbles have with each other. This allows to apply some concepts developed for the magnetic bubble technology to the skyrmion technology and gives an interesting point of view on the shortcomings that the application of skyrmions to both memory and logic. A first aspect that is really interesting is how the information is treated in the logic-in-memory case. The fact that bubble logic was not comparable in terms of speed with transistor technology was also true for this technology as it is true with skyrmions. The speed of the rotating field, needed for information movement, was indeed 100 kHz for a reliable movement and a complete cycle was required to move bubbles from one guiding element to the other. The logic-in-memory case was then, the only possible employment of the bubble logic technology. In particular, as shown in the above implementation, the memory should have had a preferable way to "pass" the information to the logic without producing a loss of the original data. The logic proposed uses a deflector to produce the data replication on another path that is continuously fed with new bubbles. A similar idea was applied in the present thesis for the duplication mechanism in chapter 6, with the AES implementation.

The duplication mechanism employs the same principle of deflection as shown in figure 9.6: the information that enters in the NOT/COPY gate is duplicated at the top and the bottom output while it is negated in the central line. This information can be used to realize logic operations while one copy of the original information can be routed back to the memory.

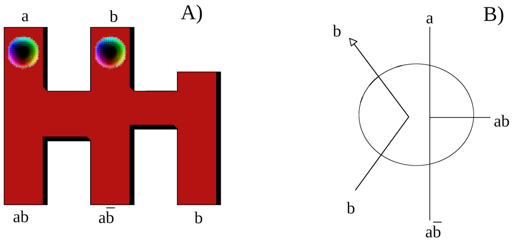


Figure 9.6: A skyrmion NOT/COPY element A) and the basic dynamical deflector B). As can be seen the NOT/COPY gate is equivalent to a dynamical deflector for bubble logic

In the same way many the concept on the completeness and on the optimal number of generators elements for the realization of a complete logic. In [141] the authors proof that with an element of the category **IB** with a minimum of 2 generators is complete. The element **IB** produce three outputs as shown in figure 9.7. This idea still stand also for skyrmion technology. The element **IB** is in fact the NOT/COPY gate cited above.

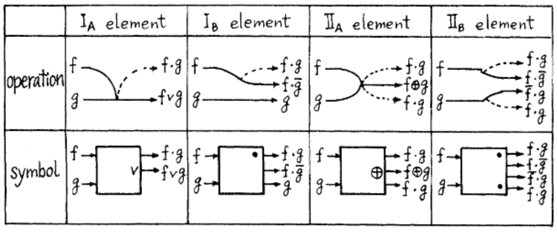


Figure 9.7: Classification of different logic elements in bubble technology. Image reproduced from [141]

The above observations show how different concepts of bubble memories are still valid for skyrmion technology. Moreover having similar interactions also some proposals for logic are applicable to the skyrmion case.

# Bibliography

[1] William Fuller Brown Jr. “Micromagnetics, domains, and resonance”. In: *Journal of Applied Physics* 30.4 (1959), S62–S69.

[2] LD Landau and EM Lifshitz. *On the Theory of the Dispersion of Magnetic Permeability in Ferromagnetic Bodies. Reproduced in Collected Papers of LD Landau.* 1935.

[3] Thomas L Gilbert. “A Lagrangian formulation of the gyromagnetic equation of the magnetization field”. In: *Phys. Rev.* 100 (1955), p. 1243.

[4] Giorgio Bertotti. *Hysteresis in magnetism: for physicists, materials scientists, and engineers.* Gulf Professional Publishing, 1998.

[5] Hans T Nembach et al. “Linear relation between Heisenberg exchange and interfacial Dzyaloshinskii–Moriya interaction in metal films”. In: *Nature Physics* 11.10 (2015), pp. 825–829.

[6] C Pappas et al. “Chiral paramagnetic skyrmion-like phase in MnSi”. In: *Physical review letters* 102.19 (2009), p. 197202.

[7] SX Huang and CL Chien. “Extended skyrmion phase in epitaxial FeGe (111) thin films”. In: *Physical review letters* 108.26 (2012), p. 267201.

[8] Wanjun Jiang et al. “Direct observation of the skyrmion Hall effect”. In: *Nature Physics* 13.2 (2017), pp. 162–169.

[9] Samridh Jaiswal et al. “Investigation of the Dzyaloshinskii-Moriya interaction and room temperature skyrmions in W/CoFeB/MgO thin films and microwires”. In: *Applied Physics Letters* 111.2 (2017), p. 022409.

[10] Shawn D Pollard et al. “Observation of stable Néel skyrmions in cobalt/-palladium multilayers with Lorentz transmission electron microscopy”. In: *Nature communications* 8.1 (2017), pp. 1–8.

[11] M Grimsditch, Y Jaccard, and Ivan K Schuller. “Magnetic anisotropies in dot arrays: Shape anisotropy versus coupling”. In: *Physical Review B* 58.17 (1998), p. 11539.

[12] John MD Coey. *Magnetism and magnetic materials.* Cambridge university press, 2010.

[13] Stephan Breitkreutz-von Gamm. "Perpendicular Nanomagnetic Logic: Digital Logic Circuits from Field-coupled Magnets". Dissertation. München: Technische Universität München, 2015.

[14] Günther Bayreuther et al. "Magnetocrystalline volume and interface anisotropies in epitaxial films: Universal relation and Néel's model". In: *Journal of applied physics* 93.10 (2003), pp. 8230–8235.

[15] Jia Zhang et al. "Model of orbital populations for voltage-controlled magnetic anisotropy in transition-metal thin films". In: *Physical Review B* 96.1 (2017), p. 014435.

[16] Pedram Khalili Amiri and Kang L Wang. "Voltage-controlled magnetic anisotropy in spintronic devices". In: *Spin*. Vol. 2. 03. World Scientific. 2012, p. 1240002.

[17] Takayuki Nozaki et al. "Recent progress in the voltage-controlled magnetic anisotropy effect and the challenges faced in developing voltage-torque MRAM". In: *Micromachines* 10.5 (2019), p. 327.

[18] William Fuller Brown Jr. "Thermal fluctuations of a single-domain particle". In: *Physical review* 130.5 (1963), p. 1677.

[19] Arne Vansteenkiste et al. "The design and verification of Mumax3". In: *AIP Advances* 4.10 (2014), p. 107133. DOI: 10.1063/1.4899186. URL: http://doi.org/10.1063/1.4899186.

[20] Marc Benjamin Hahn. "Temperature in micromagnetism: cell size and scaling effects of the stochastic Landau–Lifshitz equation". In: *Journal of Physics Communications* 3.7 (2019), p. 075009.

[21] Albert Fert, Vincent Cros, and Joao Sampaio. "Skyrmions on the track". In: *Nature nanotechnology* 8.3 (2013), pp. 152–156.

[22] Ulrich K Roessler, AN Bogdanov, and C Pfleiderer. "Spontaneous skyrmion ground states in magnetic metals". In: *Nature* 442.7104 (2006), pp. 797–801.

[23] Sebastian Mühlbauer et al. "Skyrmion lattice in a chiral magnet". In: *Science* 323.5916 (2009), pp. 915–919.

[24] Wanjun Jiang et al. "Direct observation of the skyrmion Hall effect". In: *Nature Physics* 13.2 (2017), pp. 162–169.

[25] AV Ognev et al. "Magnetic Direct-Write Skyrmion Nanolithography". In: *ACS nano* 14.11 (2020), pp. 14960–14970.

[26] Felix Büttner, Ivan Lemesh, and Geoffrey SD Beach. "Theory of isolated magnetic skyrmions: From fundamentals to room temperature applications". In: *Scientific reports* 8.1 (2018), pp. 1–12.

[27] Naoto Nagaosa and Yoshinori Tokura. "Topological properties and dynamics of magnetic skyrmions". In: *Nature nanotechnology* 8.12 (2013), pp. 899–911.

[28] Dhritiman Bhattacharya et al. “Creation and annihilation of non-volatile fixed magnetic skyrmions using voltage control of magnetic anisotropy”. In: *Nature Electronics* 3.9 (2020), pp. 539–545.

[29] Kayla Fallon et al. “Controlled individual skyrmion nucleation at artificial defects formed by ion irradiation”. In: *Small* 16.13 (2020), p. 1907450.

[30] Simone Finizio et al. “Deterministic field-free skyrmion nucleation at a nanoengineered injector device”. In: *Nano letters* 19.10 (2019), pp. 7246–7255.

[31] João Sampaio et al. “Nucleation, stability and current-induced motion of isolated magnetic skyrmions in nanostructures”. In: *Nature nanotechnology* 8.11 (2013), pp. 839–844.

[32] Philipp Dürrenfeld et al. “Controlled skyrmion nucleation in extended magnetic layers using a nanocontact geometry”. In: *Physical Review B* 96.5 (2017), p. 054430.

[33] Yan Zhou and Motohiko Ezawa. “A reversible conversion between a skyrmion and a domain-wall pair in a junction geometry”. In: *Nature communications* 5.1 (2014), pp. 1–8.

[34] Wanjun Jiang et al. “Blowing magnetic skyrmion bubbles”. In: *Science* 349.6245 (2015), pp. 283–286.

[35] Junichi Iwasaki, Masahito Mochizuki, and Naoto Nagaosa. “Current-induced skyrmion dynamics in constricted geometries”. In: *Nature nanotechnology* 8.10 (2013), pp. 742–747.

[36] Arianna Casiraghi et al. “Individual skyrmion manipulation by local magnetic field gradients”. In: *Communications Physics* 2.1 (2019), pp. 1–9.

[37] Marine Schott et al. “The skyrmion switch: turning magnetic skyrmion bubbles on and off with an electric field”. In: *Nano letters* 17.5 (2017), pp. 3006–3012.

[38] Wang Kang et al. “Voltage controlled magnetic skyrmion motion for racetrack memory”. In: *Scientific reports* 6.1 (2016), pp. 1–11.

[39] J Stöhr and Y Wu. “X-ray magnetic circular dichroism: Basic concepts and theory for 3d transition metal atoms”. In: *New Directions in Research with Third-Generation Soft X-Ray Synchrotron Radiation Sources*. Springer, 1994, pp. 221–250.

[40] Seonghoon Woo et al. “Observation of room-temperature magnetic skyrmions and their current-driven dynamics in ultrathin metallic ferromagnets”. In: *Nature materials* 15.5 (2016), pp. 501–506.

[41] Roméo Juge et al. “Magnetic skyrmions in confined geometries: Effect of the magnetic field and the disorder”. In: *Journal of Magnetism and Magnetic Materials* 455 (2018), pp. 3–8.

[42] Constance Moreau-Luchaire et al. “Additive interfacial chiral interaction in multilayers for stabilization of small individual skyrmions at room temperature”. In: *Nature nanotechnology* 11.5 (2016), pp. 444–448.

[43] Keita Hamamoto, Motohiko Ezawa, and Naoto Nagaosa. “Purely electrical detection of a skyrmion in constricted geometry”. In: *Applied Physics Letters* 108.11 (2016), p. 112401.

[44] Davide Maccariello et al. “Electrical detection of single magnetic skyrmions in metallic multilayers at room temperature”. In: *Nature nanotechnology* 13.3 (2018), pp. 233–237.

[45] Riccardo Tomasello et al. “Electrical detection of single magnetic skyrmion at room temperature”. In: *Aip Advances* 7.5 (2017), p. 056022.

[46] NE Penthorn et al. “Experimental observation of single skyrmion signatures in a magnetic tunnel junction”. In: *Physical review letters* 122.25 (2019), p. 257201.

[47] Shasha Wang et al. “Electrical Detection of Magnetic Skyrmions”. In: *Journal of Low Temperature Physics* 197.3 (2019), pp. 321–336.

[48] Stuart S. P. Parkin, Masamitsu Hayashi, and Luc Thomas. “Magnetic Domain-Wall Racetrack Memory”. In: *Science* 320.5873 (2008), pp. 190–194. ISSN: 0036-8075. DOI: 10.1126/science.1145799. eprint: https://science.sciencemag.org/content/320/5873/190.full.pdf. URL: https://science.sciencemag.org/content/320/5873/190.

[49] Giovanni Finocchio et al. “Magnetic skyrmions: from fundamental to applications”. In: *Journal of Physics D: Applied Physics* 49.42 (2016), p. 423001.

[50] Kai Litzius et al. “Skyrmion Hall effect revealed by direct time-resolved X-ray microscopy”. In: *Nature Physics* 13.2 (2017), pp. 170–175.

[51] AA Thiele. “Steady-state motion of magnetic domains”. In: *Physical Review Letters* 30.6 (1973), p. 230.

[52] Riccardo Tomasello et al. “A strategy for the design of skyrmion racetrack memories”. In: *Scientific reports* 4 (2014), p. 6784.

[53] Jeroen Mulkers, Bartel Van Waeyenberge, and Milorad V. Milošević. “Effects of spatially-engineered Dzyaloshinskii-Moriya interaction in ferromagnetic films”. In: *Physical Review B* 95.14 (2017), p. 144401. DOI: 10.1103/PhysRevB.95.144401. URL: doi.org/10.1103/PhysRevB.95.144401.

[54] M.J. Donahue and D.G. Porter. “OOMMF User’s Guide, Version 1.0”. In: *Interagency Report NISTIR 6376, National Institute of Standards and Technology, Gaithersburg, MD* (Sept 1999).

[55] Maverick Chauwin et al. "Skyrmion Logic System for Large-Scale Reversible Computation". In: *Phys. Rev. Applied* 12 (6 Dec. 2019), p. 064053. DOI: 10.1103/PhysRevApplied.12.064053. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.12.064053.

[56] Xichao Zhang, Motohiko Ezawa, and Yan Zhou. "Magnetic skyrmion logic gates: conversion, duplication and merging of skyrmions". In: *Scientific reports* 5.1 (2015), pp. 1–8.

[57] Shijiang Luo et al. "Reconfigurable skyrmion logic gates". In: *Nano letters* 18.2 (2018), pp. 1180–1184.

[58] M. G. Mankalale et al. "SkyLogic—A Proposal for a Skyrmion-Based Logic Device". In: *IEEE Transactions on Electron Devices* 66.4 (2019), pp. 1990–1996. DOI: 10.1109/TED.2019.2899263.

[59] Wang Kang et al. "Voltage controlled magnetic skyrmion motion for racetrack memory". In: *Scientific reports* 6.1 (2016), pp. 1–11.

[60] Gyoergy Csaba, Markus Becherer, and Wolfgang Porod. "Development of CAD tools for nanomagnetic logic devices". In: *International Journal of Circuit Theory and Applications* 41.6 (2013), pp. 634–645.

[61] Pramey Upadhyaya et al. "Electric-field guiding of magnetic skyrmions". In: *Physical Review B* 92.13 (2015), p. 134411.

[62] Benjamin W Walker et al. "Skyrmion Logic Clocked via Voltage Controlled Magnetic Anisotropy". In: *arXiv preprint arXiv:2103.02724* (2021).

[63] C. Zhang et al. "Critical role of W deposition condition on spin-orbit torque induced magnetization switching in nanoscale W/CoFeB/MgO". In: *Applied Physics Letters* 109.19 (2016), p. 192405. DOI: 10.1063/1.4967475. eprint: https://doi.org/10.1063/1.4967475. URL: https://doi.org/10.1063/1.4967475.

[64] Huan Liu. "Routing table compaction in ternary CAM". In: *IEEE Micro* 22.1 (2002), pp. 58–64.

[65] Satendra Kumar Maurya and Lawrence T Clark. "A dynamic longest prefix matching content addressable memory for IP routing". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 19.6 (2010), pp. 963–972.

[66] Atsushi Ooka et al. "Design of a high-speed content-centric-networking router using content addressable memory". In: *2014 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE. 2014, pp. 458–463.

[67] Le Zheng, Sangho Shin, and Sung-Mo Steve Kang. "Memristor-based ternary content addressable memory (mTCAM) for data-intensive computing". In: *Semiconductor Science and Technology* 29.10 (2014), p. 104010.

[68] Qing Guo et al. "Resistive ternary content addressable memory systems for data-intensive computing". In: *IEEE Micro* 35.5 (2015), pp. 62–71.

[69] Pilin Junsangsri and Fabrizio Lombardi. "A memristor-based TCAM (ternary content addressable memory) cell: design and evaluation". In: *Proceedings of the great lakes symposium on VLSI*. 2012, pp. 311–314.

[70] Wei Xu, Tong Zhang, and Yiran Chen. "Design of spin-torque transfer magnetoresistive RAM and CAM/TCAM with high sensing and search speed". In: *IEEE transactions on very large scale integration (VLSI) systems* 18.1 (2009), pp. 66–74.

[71] Xiao Yang, Wanlong Chen, and Frank Z Wang. "A memristor-cam (content addressable memory) cell: New design and evaluation". In: *International conference on computer science and information technology*. 2013, pp. 1045–1048.

[72] Yong Ye et al. "CAM-based retention-aware DRAM (CRA-DRAM) for refresh power reduction". In: *IEICE Electronics Express* 14.10 (2017), pp. 20170053–20170053.

[73] Hiu Tung Fook et al. "Mitigation of magnus force in current-induced skyrmion dynamics". In: *IEEE Transactions on Magnetics* 51.11 (2015), pp. 1–4.

[74] Indra Purnama et al. "Guided current-induced skyrmion motion in 1D potential well". In: *scientific Reports* 5.1 (2015), pp. 1–9.

[75] Dustin M Lattery et al. "Low Gilbert damping constant in perpendicularly magnetized W/CoFeB/MgO films with high thermal stability". In: *Scientific reports* 8.1 (2018), pp. 1–9.

[76] Dhritiman Bhattacharya et al. "Creation and annihilation of non-volatile fixed magnetic skyrmions using voltage control of magnetic anisotropy". In: *Nature Electronics* 3.9 (2020), pp. 539–545.

[77] Telajala Venkata Mahendra, Sandeep Mishra, and Anup Dandapat. "Self-controlled high-performance precharge-free content-addressable memory". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 25.8 (2017), pp. 2388–2392.

[78] Kostas Pagiamtzis and Ali Sheikholeslami. "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey". In: *IEEE journal of solid-state circuits* 41.3 (2006), pp. 712–727.

[79] Yen-Jen Chang and Yuan-Hong Liao. "Hybrid-type CAM design for both power and performance efficiency". In: *IEEE transactions on very large scale integration (VLSI) systems* 16.8 (2008), pp. 965–974.

[80] Diptesh Datta et al. "A 10T, 0.25 fJ/bit/search Mixed-VT Pseudo Precharge-Free Content Addressable Memory". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* (2021).

[81] Shoun Matsunaga et al. "Design of a nine-transistor/two-magnetic-tunnel-junction-cell-based low-energy nonvolatile ternary content-addressable memory". In: *Japanese Journal of Applied Physics* 51.2S (2012), 02BM06.

[82] Byungkyu Song et al. "A 10T-4MTJ nonvolatile ternary CAM cell for reliable search operation and a compact area". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 64.6 (2016), pp. 700–704.

[83] Chengzhi Wang et al. "A novel MTJ-based non-volatile ternary content-addressable memory for high-speed, low-power, and high-reliable search operation". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 66.4 (2018), pp. 1454–1464.

[84] M. Vacca et al. "Logic-In-Memory Architecture For Min/Max Search". In: *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. 2018, pp. 853–856.

[85] Luca Gnoli et al. "Skyrmion Logic-In-Memory Architecture for Maximum/Minimum Search". In: *Electronics* 10.2 (2021), p. 155.

[86] *ElmerFEM, Open source multiphysical simulation software.* http://www.elmerfem.org/.

[87] Z. Zhang et al. "Skyrmion-Based Ultra-Low Power Electric-Field-Controlled Reconfigurable (SUPER) Logic Gate". In: *IEEE Electron Device Letters* 40.12 (2019), pp. 1984–1987. DOI: 10.1109/LED.2019.2946863.

[88] Joan Daemen and Vincent Rijmen. "AES proposal: Rijndael". In: (1999).

[89] Christof Paar and Jan Pelzl. *Understanding cryptography: a textbook for students and practitioners.* Springer Science & Business Media, 2009.

[90] Yiqun Zhang et al. "A compact 446 Gbps/W AES accelerator for mobile SoC and IoT in 40nm". In: *2016 IEEE symposium on VLSI circuits (VLSI-circuits)*. IEEE. 2016, pp. 1–2.

[91] Mimi Xie et al. "Securing emerging nonvolatile main memory with fast and energy-efficient AES in-memory implementation". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26.11 (2018), pp. 2443–2455.

[92] Shunsuke Fukami et al. "Domain wall motion device for nonvolatile memory and logic—Size dependence of device properties". In: *IEEE Transactions on Magnetics* 50.11 (2014), pp. 1–6.

[93] Shaahin Angizi et al. “Design and evaluation of a spintronic in-memory processing platform for nonvolatile data encryption”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37.9 (2017), pp. 1788–1801.

[94] Z Abid et al. “Efficient cmol gate designs for cryptography applications”. In: *IEEE transactions on nanotechnology* 8.3 (2009), pp. 315–321.

[95] Yuhao Wang et al. “DW-AES: A domain-wall nanowire-based AES for high throughput and energy-efficient data encryption in non-volatile memory”. In: *IEEE Transactions on Information Forensics and Security* 11.11 (2016), pp. 2426–2440.

[96] Shaodi Wang et al. “Comparative Evaluation of Spin-Transfer-Torque and Magnetoelectric Random Access Memory”. In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 6.2 (2016), pp. 134–145. DOI: 10.1109/JETCAS.2016.2547681.

[97] M. Chen et al. “Cache Memory Design With Magnetic Skyrmions in a Long Nanotrack”. In: *IEEE Transactions on Magnetics* 55.8 (2019), pp. 1–9. DOI: 10.1109/TMAG.2019.2909188.

[98] Giulia Santoro, Giovanna Turvani, and Mariagrazia Graziano. “New logic-in-memory paradigms: An architectural and technological perspective”. In: *Micromachines* 10.6 (2019), p. 368.

[99] Roméo Juge et al. “Helium ions put magnetic skyrmions on the track”. In: *Nano Letters* (2021).

[100] Craig S Lent et al. “Quantum cellular automata”. In: *Nanotechnology* 4.1 (1993), p. 49.

[101] Yuhui Lu and Craig S Lent. “Theoretical study of molecular quantum-dot cellular automata”. In: *Journal of Computational Electronics* 4.1 (2005), pp. 115–118.

[102] Yuhui Lu, Mo Liu, and Craig Lent. “Molecular quantum-dot cellular automata: From molecular structure to circuit dynamics”. In: *Journal of applied physics* 102.3 (2007), p. 034311.

[103] Craig S Lent, Beth Isaksen, and Marya Lieberman. “Molecular quantum-dot cellular automata”. In: *Journal of the American Chemical Society* 125.4 (2003), pp. 1056–1063.

[104] Alexandra Imre et al. “Majority logic gate for magnetic quantum-dot cellular automata”. In: *Science* 311.5758 (2006), pp. 205–208.

[105] RP Cowburn and ME Welland. “Room temperature magnetic quantum cellular automata”. In: *Science* 287.5457 (2000), pp. 1466–1468.

[106] Michael T Niemier, György Csaba, and Wolfgang Porod. “Nanomagnet Logic: A Magnetic Implementation of Quantum-dot Cellular Automata”. In: *Emerging Nanoelectronic Devices*. Wiley Online Library, 2014, pp. 417–442.

[107] Mohmmad Tanvir Alam et al. “On-Chip Clocking for Nanomagnet Logic Devices”. In: *IEEE Transactions on Nanotechnology* 9.3 (2010), pp. 348–351. DOI: 10.1109/TNANO.2010.2041248.

[108] Marco Vacca et al. “Magnetoelastic Clock System for Nanomagnet Logic”. In: *IEEE Transactions on Nanotechnology* 13.5 (2014), pp. 963–973. DOI: 10.1109/TNANO.2014.2333657.

[109] Marco Vacca et al. “Electric Clock for NanoMagnet Logic Circuits”. In: *Field-Coupled Nanocomputing: Paradigms, Progress, and Perspectives*. Ed. by Neal G. Anderson and Sanjukta Bhanja. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 73–110. ISBN: 978-3-662-43722-3. DOI: 10.1007/978-3-662-43722-3_5. URL: https://doi.org/10.1007/978-3-662-43722-3_5.

[110] Mariagrazia Graziano et al. “An NCL-HDL Snake-Clock-Based Magnetic QCA Architecture”. In: *IEEE Transactions on Nanotechnology* 10.5 (2011), pp. 1141–1149. DOI: 10.1109/TNANO.2011.2118229.

[111] David Carlton et al. “Investigation of Defects and Errors in Nanomagnetic Logic Circuits”. In: *IEEE Transactions on Nanotechnology* 11.4 (2012), pp. 760–762. DOI: 10.1109/TNANO.2012.2196445.

[112] Baojun Liu et al. “On-chip readout circuit for nanomagnetic logic”. In: *IET Circuits, Devices & Systems* 8.1 (2014), pp. 65–72.

[113] MT Niemier et al. “Nanomagnet logic: progress toward system-level integration”. In: *Journal of Physics: Condensed Matter* 23.49 (2011), p. 493202.

[114] Markus Becherer et al. “Nanomagnetic logic clocked in the mhz regime”. In: *2013 Proceedings of the European Solid-State Device Research Conference (ESSDERC)*. IEEE. 2013, pp. 276–279.

[115] Stephan Breitkreutz et al. “1-bit full adder in perpendicular nanomagnetic logic using a novel 5-input majority gate”. In: *EPJ web of conferences*. Vol. 75. EDP Sciences. 2014, p. 05001.

[116] M Becherer et al. “Towards on-chip clocking of perpendicular nanomagnetic logic”. In: *Solid-State Electronics* 102 (2014), pp. 46–51.

[117] Debanjan Bhowmik, Long You, and Sayeef Salahuddin. “Spin Hall effect clocking of nanomagnetic logic without a magnetic field”. In: *Nature nanotechnology* 9.1 (2014), pp. 59–63.

[118] Adam Papp et al. “Threshold gate-based circuits from nanomagnetic logic”. In: *IEEE Transactions on Nanotechnology* 13.5 (2014), pp. 990–996.

[119] G Turvani et al. “A compact physical model for the simulation of pNML-based architectures”. In: *AIP Advances* 7.5 (2017), p. 056005.

[120] F. Riente et al. “MagCAD: Tool for the Design of 3-D Magnetic Circuits”. In: *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits* 3 (2017), pp. 65–73. DOI: 10.1109/JXCDC.2017.2756981.

[121] F Riente et al. “Ta/CoFeB/MgO analysis for low power nanomagnetic devices”. In: *AIP Advances* 10.12 (2020), p. 125229.

[122] Simon Mendisch et al. “Controlling Domain-Wall Nucleation in Ta/CoFeB/MgO Nanomagnets via Local Ga+ Ion Irradiation”. In: *arXiv preprint arXiv:2102.07540* (2021).

[123] Luc Thomas et al. “Racetrack memory: a high-performance, low-cost, non-volatile memory based on magnetic domain walls”. In: *2011 International Electron Devices Meeting*. IEEE. 2011, pp. 24–2.

[124] Kwang-Su Ryu et al. “Current induced tilting of domain walls in high velocity motion along perpendicularly magnetized micron-sized Co/Ni/Co racetracks”. In: *Applied Physics Express* 5.9 (2012), p. 093006.

[125] Stuart Parkin and See-Hun Yang. “Memory on the racetrack”. In: *Nature nanotechnology* 10.3 (2015), pp. 195–198.

[126] Xiaoxuan Zhao et al. “Ultra-efficient spin–orbit torque induced magnetic switching in W/CoFeB/MgO structures”. In: *Nanotechnology* 30.33 (2019), p. 335707.

[127] Markus Becherer et al. “A monolithic 3D integrated nanomagnetic co-processing unit”. In: *Solid-State Electronics* 115 (2016), pp. 74–80.

[128] Fabrizio Riente et al. “Towards logic-in-memory circuits using 3d-integrated nanomagnetic logic”. In: *2016 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE. 2016, pp. 1–8.

[129] Felix Büttner, Ivan Lemesh, and Geoffrey SD Beach. “Theory of isolated magnetic skyrmions: From fundamentals to room temperature applications”. In: *Scientific reports* 8.1 (2018), pp. 1–12.

[130] JL Archer. “Megabit bubble chip”. In: *Digests of the Intermag Conference*. 1977, pp. 11–11.

[131] W Kinsner. “ACTUARIAL RESEARCH CLEARING HOUSE 1982 VOL.” In: ().

[132] H Banks. “The computer bubble that burst”. In: *The New York Times* (1981), 4F.

[133] C Kooy. “Experimental and theoretical study of the domain configuration in thin layers of BaFe_< 12> O_< 19”. In: *Philips Res. Repts* 15.7 (1960).

[134] AH Bobeck. “Properties and device applications of magnetic domains in orthoferrites”. In: *The bell system technical journal* 46.8 (1967), pp. 1901–1925.

[135] Andrew H Bobeck and HED Scovil. “Magnetic bubbles”. In: *Scientific American* 224.6 (1971), pp. 78–91.

[136] Tien Chi Chen and Hsu Chang. “Magnetic bubble memory and logic”. In: *Advances in Computers.* Vol. 17. Elsevier, 1978, pp. 223–282.

[137] TC Chen, H Chang, and SY Lee. “A magnetic-bubble pipelined rewritable universal logic array”. In: *Digest 1975 Intermag Conf.* 1975, pp. 25–9.

[138] Robert C Minnick et al. “Magnetic bubble computer systems”. In: *Proceedings of the December 5-7, 1972, fall joint computer conference, part II.* 1972, pp. 1279–1298.

[139] Richard P. Williams. “Serial integer arithmetic with magnetic bubbles”. In: *IEEE Transactions on Computers* 26.3 (1977), pp. 260–264.

[140] Share Young Lee and Hsu Chang. “Associative-search bubble devices for content-addressable memory and array logic”. In: *IEEE Transactions on Computers* 100.9 (1979), pp. 627–636.

[141] Kozo Kinoshita, Tsutomu Sasao, and Jun Matsuda. “On magnetic bubble logic circuits”. In: *IEEE Transactions on Computers* 100.3 (1976), pp. 247–253.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.