POLITECNICO DI TORINO

SCUOLA DI DOTTORATO
Dottorato in Ingegneria Elettronica e delle Comunicazioni – XXV ciclo

Tesi di Dottorato

# Emerging Technologies - NanoMagnets Logic (NML)


**Marco Vacca**

**Tutore**
prof. Mariagrazia Graziano

**Coordinatore del corso di dottorato**
prof. Ivo Montrosset

April 2013

# Summary

In the last decades CMOS technology has ruled the electronic scenario thanks to the constant scaling of transistor sizes. With the reduction of transistor sizes circuit area decreases, clock frequency increases and power consumption decreases accordingly. However CMOS scaling is now approaching its physical limits and many believe that CMOS technology will not be able to reach the end of the Roadmap. This is mainly due to increasing difficulties in the fabrication process, that is becoming very expensive, and to the unavoidable impact of leakage losses, particularly thanks to gate tunnel current.

In this scenario many alternative technologies are studied to overcome the limitations of CMOS transistors. Among these possibilities, magnetic based technologies, like NanoMagnet Logic (NML) are among the most interesting. The reason of this interest lies in their magnetic nature, that opens up entire new possibilities in the design of logic circuits, like the possibility to mix logic and memory in the same device. Moreover they have no standby power consumption and potentially a much lower power consumption of CMOS transistors.

In literature NML logic is well studied and theoretical and experimental proofs of concept were already found. However two important points are not enough considered in the analysis approach followed by most of the work in literature. First of all, no complex circuits are analyzed. NML logic is very different from CMOS technologies, so to completely understand the potential of this technology it is mandatory to investigate complex architectures. Secondly, most of the solutions proposed do not take into account the constraints derived from fabrication process, making them unrealistic and difficult to be fabricated experimentally.

This thesis focuses therefore on NML logic keeping into account these two important limitations in the research approach followed in literature. The aim is to obtain a complete and accurate overview of NML logic, finding realistic circuital solutions and trying to improve at the same time their performance. After a brief and complete introduction (Chapter 1), the thesis is divided in two parts, which cover the two fundamental points followed in this three years of research: A circuits architecture analysis and a technological analysis.

In the architecture analysis first an innovative VHDL model is described in Chapter 2. This model is extensively used in the analysis because it allows fast simulation of complex circuits, with, at the same time, the possibility to estimate circuit performance, like area and power consumption. In Chapter 3 the problem of signals synchronization in complex NML circuits is analyzed and solved, using as benchmark a simple but complete NML microprocessor. Different solutions based on asynchronous logic are studied and a new asynchronous solution, specifically designed to exploit the potential of NML logic, is developed. In Chapter 4 the layout of NML circuits is studied on a more physical level, considering the limitations of fabrication processes. The layout of NML circuits is therefore changed accordingly to these constraints. Secondly CMOS circuits architectures are compared to more simple architectures, evaluating therefore which one is more suited for NML logic. Finally the problem of interconnections in NML technology is analyzed and solutions to improve it are found. In Chapter 5 the problem of feedback signals in heavy pipelined technologies, like NML, is studied. Solutions to improve performances and synchronize signals are developed. Systolic arrays are then analyzed as possible candidate to exploit NML potential. Finally in Chapter 6 ToPoliNano, a simulator dedicated to NML and other emerging technologies, that we are developing, is described. This simulator allows to follow the same top-down approach followed for CMOS technology. The layout generator and the simulation engine are detailed described.

In the first chapter of the technological analysis (Chapter 7), the performance of NML logic is explored throughout low level simulations. The aim is to understand if these circuits can be fabricated with optical lithography, allowing therefore the commercial development of NML logic. Basic logic gates and the clock system are there analyzed from a low level perspective. In Chapter 8 an innovative electric clock system for NML technology is shown and the first experimental results are reported. This clock system allows to achieve true low power for NML technology, obtaining a reduction of power consumption of 20 times considering the best CMOS transistors available. This power consumption takes into account all the losses, also the clock system losses. Moreover the solution presented can be fabricated with current technological processes.

The research work behind this thesis represents an important breakthrough in NML logic. The solutions here presented allow the design and fabrication of complex NML circuits, considering the particular characteristics of this technology and considerably improving the performance. Moreover the technological solutions here presented allow the design and fabrication of circuits with available fabrication process with a considerable advantage over CMOS in terms of power consumption. This thesis represents therefore a considerable step froward in the study and development of NML technology.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Quantum dot Cellular Automata (QCA)

In recent years the original cellular automata principle [8] has been used to develop Quantum dot Cellular Automata (QCA) technology [9],[10]. In this technology identical square shape cells encode logic values ('0' and '1') using bistable charge configurations [11]. The base cell, shown in Figure 1.1.A, is constituted by four quantum dots, one on each corner. Each quantum dot can be filled with electrons. Since electrons repel each others at the equilibrium only the two diagonal dots will result occupied. There are only two diagonals, therefore only two states are possible, which therefore represent the two logic values '0' and '1'.

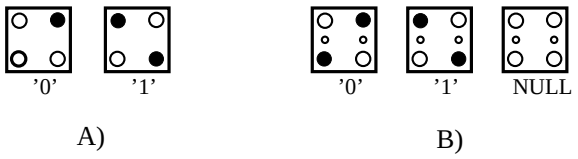



Figure 1.1. Quantum dot Cellular Automata (QCA) cells. A) Four dots cells. B) Six dots cells.

To build circuits QCA cells are placed on a plane near each other [12]. Information propagates through the circuit thanks to electrostatic interaction. The simplest circuit is the wire, that is shown in Figure 1.2. Figure 1.2.A shows the initial state of the wire. When the first cell is externally forced from '0' to '1' using, for example, an electric field (Figure 1.2.B), the second cell will switch due to the electrostatic interaction between adjacent electrons (Figure 1.2.C). Finally the last cell will switch (Figure 1.2.D). As it is possible to see information propagates through the circuit with a Domino-like effect.

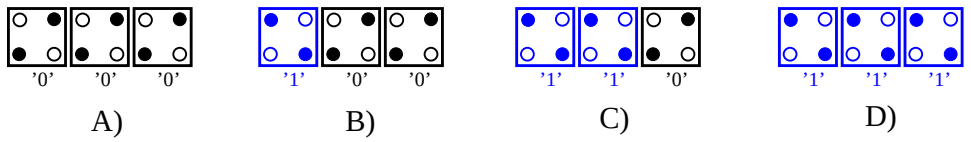



Figure 1.2. Quantum dot Cellular Automata (QCA) wire. A) Starting condition. B) Input cell is forced to 1. C) Second cell switches to 1, due to the electrostatic interaction. D) Third cell switches to 1.

There are four basic logic gates in QCA technology (Fig. 1.3): The wire (Fig. 1.3.A), the inverter (Fig. 1.3.B), the majority gate (Fig. 1.3.C) and the crosswire (Fig. 1.3.D). While in CMOS technology wires are classified as simple interconnections, in QCA the wire is a logic gate, because it is equal to a chain of buffers. However to perform logic computation it is necessary to use the other two gates, the majority voter and the inverter. While the inverter performs a simple signal inversion, the logic equation of the second one is uncommon, at least in CMOS technology: The value of the output is equal to the value of the majority of the inputs. Finally the crosswire is a block that represent a special characteristic of this technology: Thanks to this block two wires can be crossed on the same plane without interference. This allows the construction of logic circuits on one plane, reducing the fabrication complexity.

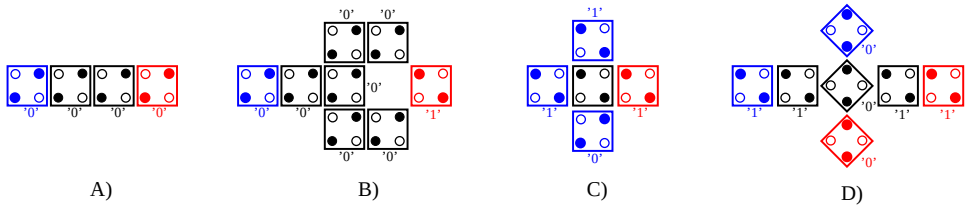



Figure 1.3. Quantum dot Cellular Automata (QCA) basic blocks. A) Wire. B) Inverter. C) Majority gate. D) Crosswire.

While this principle works in theory, practically the electrostatic interaction between neighbor cells is not strong enough to force switching in a neighbor cell because the energy barrier between different states is very high. A second important limitation is that only a finite number of cells can be cascaded. If too many cells are cascaded there will be errors due to thermal noise or other noise sources. As a consequence the so-called clock [13], an external means that allows to control the information flux, was introduced. To use a clock mechanism the basic cell must be modified introducing two more dots (Figure 1.1.B). Applying an external electric field the potential barrier between the two stable states is lowered, therefore electrons are forced inside the two extra dots, which is an unstable state, called NULL.

Removing the field, cell switches to '0' or '1', depending on the value of neighbor cells. A spatial flow control system is introduced because only circuits composed by a limited number of cells can work without error propagation. The circuit is divided in small areas, composed by a limited number of cells, called clock zones. At every clock zone a different time varying signal is applied as shown in Figure 1.4. This allows a spatial and timing control of the circuit. In the classical clock scheme circuits are divided in four clock zones, the circuits partition and the clock signal waveforms are shown in Figure 1.4.

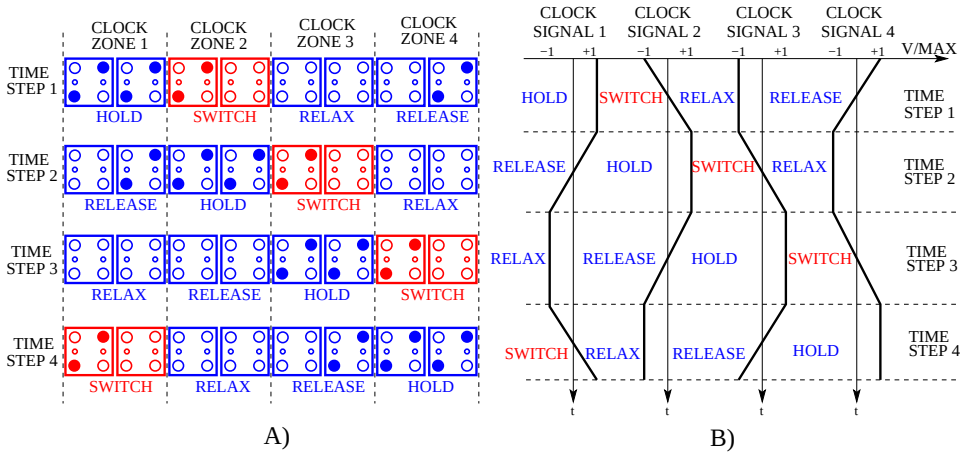



Figure 1.4. Clock mechanism. A) Clock zones. B) Clock signals.

When the clock signal is high (V = VH) the potential barrier between the two logic states is risen and therefore the cell switch is impossible. In this case the cells are in HOLD state. When the clock signal decreases from VH to VL the potential barrier decreases its value slowly, cells start to switch from a stable state to an unstable one. Cells are in the RELEASE phase. When the clock signal is low (V = VL) the potential barrier is zero, the two logic states are not separated. Cells are in the RELAX state. Finally when the clock signal rises from -1 to +1 the potential barrier increases its value slowly forcing cells in a stable state: cells therefore are in the SWITCH state. As clear from Figure 1.4.B, the clock signal is always identical, but applied with a different phase to other clock zones. This allows the spatial propagation of the signal through the circuit as shown in Figure 1.4.A. During the first time step the clock zone number 2 is in the switch phase, they are in an unstable state and are read to switch to one of the stable states. Cells at its left are in the hold state and act like an input, while cells on its right are in an unstable state so they have no influence, allowing the correct switching of the cells in clock zone 2. During the second time step the situation is the same, but in this case the clock zone number 3 is in the switch phase.

This technique allows a correct signal propagation in a specific direction. To allows signals propagation in every directions, as required to build any kind of complex circuit, clock zones must be arranged properly [14]. Figure 1.5 shows an example of possible clock zones layout which assures signal propagation in every direction. A fundamental requirement for the division of circuit area in clock zones, is that the local control must be perfect. This means that the applied electric field must be perfectly confined to the clock zone itself and must not interfere with neighbor clock zones.

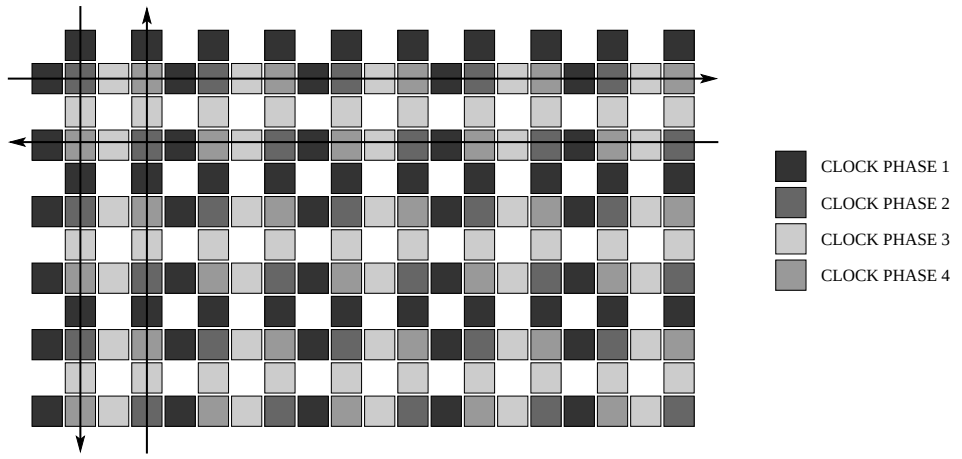



Figure 1.5. Example of complex QCA clock zones layout. *M.Graziano, M.Vacca et al."Magnetic QCA Design: Modeling, Simulation and Circuits", Cellular Automata Innovative Modelling For Science And Engineering, Intechweb.org, 2011*

The theoretical principle of the QCA can be implemented in different ways. Four are the proposals in literature for a real QCA implementation, which are briefly described in the following.

- **Metal QCA** [11][13]. The base cell is constituted by six metal lines, that act like quantum dots, on a substrate of silicon oxide. Metal lines are separated by tunnel-junction, that allow the electrons to exchange between neighbor dots. The charge configuration of the cell is read using a single electron transistor (SET). The cell works properly, but only at temperatures near the absolute zero. To work at room temperature the cell size must be reduced to atomic scale.

- **Semiconductor QCA** [15][16]. Complex structures of Si-Ge or GaAs are used to create quantum dots that are able to trap electrons. The operation temperature is higher than the metal QCA but is always too low for practical uses. In order to increase the operation temperature cell dimensions must be

reduced at some nanometers, but this is impossible with current technology. Moreover, one condition necessary for proper operations of QCA circuits is that every cell must be identical, but, if a so complex structure is realized with the desired resolution, the impact of defect rate caused by the fabrication process will make QCA inoperable, limiting every practical possibility of this implementation.

- **Molecular QCA** [17][18][19][20]. In this case, complex molecules with many oxide-reduction centers, that act like quantum dots, are used as base cell. Electrons can react with every center inside the molecule, changing the spatial distribution of the electric charge, and the logic value associated to it. The use of molecules bring many advantages, like that every QCA cell is identical to each others and the fact that molecules circuits can work at room temperature. However the most interesting aspect of the use of molecules is the switching speed expected: Molecular QCA have the possibility to obtain operating frequency of some THz; moreover the dimensions of such molecules are very small (a few nanometers), allowing the generation of circuits with a very high device densities. Molecular QCA are very attracting but their realization requires the ability of manipulating single molecules, which is not possible with up-to-date technology.

- **Magnetic QCA** or **NanoMagnet Logic (NML)** [2]. The base cell is a single domain nanomagnet, with only two possible magnetizations that represent the two logic value '0' and '1'. This is the second promising implementation of the QCA principle, because also in this case circuits can work at room temperature. Unfortunately the expected speed is lower not only than the molecular case, but also than CMOS circuits. However magnetic QCA have some specific advantages which make them attractive, in particular the low power consumption and the possibility to realize them with current technology: This allows to experiment and study the QCA principle so that most of the achievements can be adapted in a near future to molecular QCA, as soon as this solution becomes feasible.

## 1.2 Magnetic QCA or NanoMagnetic Logic (NML)

The idea of using of using magnets to build logic circuits is the realization of a sixty years old dream. Magnets were and are successfully used for memory application, so why not use them also for logic, obtaining circuits with both memory and logic ability? Unfortunately, sixty years ago, the technology was not ready for such an application, but today the situation is quite different, thanks to the huge advancements of fabrication techniques, like lithography processes. In the Magnetic

Quantum dot Cellular Automata (MQCA) implementation, also called NanoMagnetic Logic (NML), the basic cell is a nanoscale nanomagnets, with sizes between 50nm and 100nm. Magnetic materials are composed by magnetic domains, small areas with a uniform magnetization, and the behavior is governed by the hysteresis cycle, which represents how material magnetization (M) changes if an external magnetic field (H) is applied, see Figure 1.6.A. Reducing the size of magnets under the 100nm limit transforms the magnetic structure leaving only one domain left and changing the hysteresis cycle as shown in Figure 1.6.B. If this condition is reached every magnets can have only two stable states, thanks to magnetic shape anisotropy. This two stable states can be used to represent the logic values '0' and '1', as happens in every QCA cells. It is however important to keep sizes bigger than approximately 50nm, to avoid the so-called superparamagnetic effect, which makes the magnetization varying with thermal fluctuations. In general the energy barrier between the two stable states must be kept bigger than $30K_bT$, to obtain a good thermal stability. Shape anisotropy is a magnetic properties that, forces the magnetization of a magnet along its longer axis (called easy axis), so it is important that one side of magnets is bigger than the other (called hard axis). Shape anisotropy is related to the value of the demagnetization field, a field generated intrinsically to the magnet when it is magnetized. This field reaches its minimum along the longer axis of the materials, therefore the magnetization, at the equilibrium, tend to stay parallel to the easy axis. In NML logic the aspect ratio (the ratio between the longer and shorter side) lies in the range of 1.1-2.

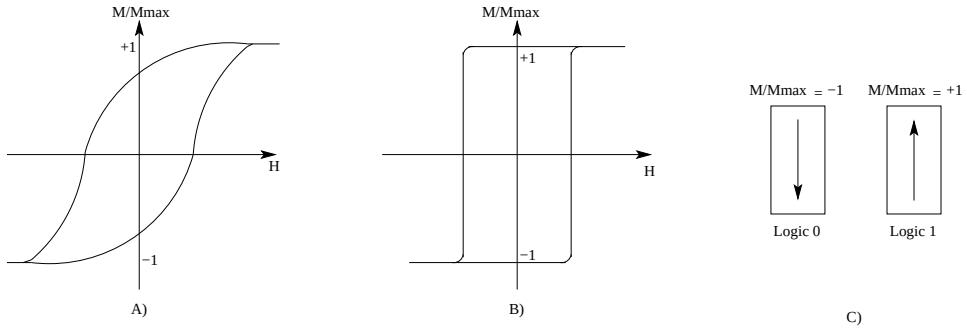



Figure 1.6. A) Multidomain magnetic material hysteresis cycle. B) Single domain magnetic material hysteresis cycle. C) Magnetic Quantum dot Cellular Automata (MQCA) cells. *M.Graziano, M.Vacca et al."Magnetic QCA Design: Modeling, Simulation and Circuits", Cellular Automata Innovative Modelling For Science And Engineering, Intechweb.org, 2011*

NanoMagnet Logic can reach a frequency between 50 MHz and 1 GHz; however they have some significant advantages, over others QCA implementations:

- they are one of the only two implementations of the QCA principle that works at room temperature;
- they can be realized with current technology, with electron beam lithography or high end optical lithography;
- they can have a very low power absorption, requiring an energy of 15-30Kb T for every nanomagnets to switch, granting the possibility to obtain very low power electronic devices (see Chapters 7 and 8 for more details on power consumption);
- they have an intrinsic memory ability, as, due to their magnetic nature they maintain the information stored also without power supply, enabling thus to define circuits with mixed computational-storage abilities;
- most of the high level research related to NML can be transposed to the molecular QCA, once technology will be ready.

### 1.2.1 Logic Gates

The logic gates available in this QCA technology are shown in Figure 1.7. They are slightly different from their equivalent in the generic QCA implementation. First of all coupling in horizontal and vertical wires is different: in horizontal wires magnets align themselves antiferromagnetically (every magnets is the inverted value of its neighbors) (Figure 1.7.A) while in vertical wires the alignment is ferromagnetic (Figure 1.7.C). As a consequence the inverter can be built simply using horizontal wires with an odd number of elements (Figure 1.7.B). The majority voter is instead the same (Figure 1.7.D). Another difference is the possibility to obtain different logic gates changing the shape of a specific magnets [6]. In this way AND gates (Figure 1.7.E) and OR gates (Figure 1.7.F) can be obtained. The crosswire is a little bit tricky to obtain and up to now not experimental evidences were already obtained [2]. Figure 1.7.G shows a possible implementation of the magnetic crosswire.

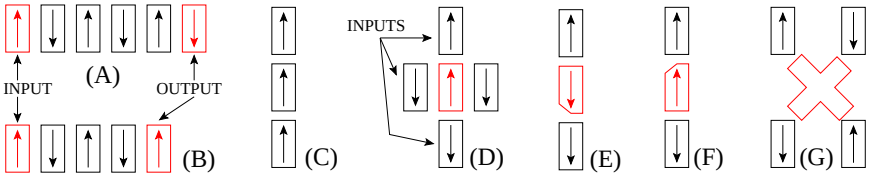



Figure 1.7. NML logic gates. A) Horizontal Wire. B) Inverter. C) Vertical Wire. D) Majority Voter. E) AND. F) OR. G) Crosswire.

### 1.2.2 Clock

It has been demonstrated (see [21]) that for NML, as well as for molecular QCA, an adiabatic switching is preferred to assure a correct information propagation and low power operations. This means that the switching of a nanomagnet from state "up" to state "down" is favored if an intermediate state is reached first. That is, similarly to what mentioned in section 1.1, an external field is applied so that the pill "memory" (previous magnetization state to "up" or "down") is erased (magnetization become a perpendicular to "up" or "down" direction), and, at this point, as soon the external field is released, an input can more easily force the new "up" or "down" magnetization to the pill (see Figure 1.8). This is particularly important when the input of nanomagnet-B is another nanomagnet-A, which can force on the coupled nanomagnet-B only a limited magnetic field due to its intrinsic characteristics (shape and material). Such external field is meant as a clock, as it is iteratively switched on and off and allows the evaluation phase, even though it has not the "traditional" function of a clock signal.

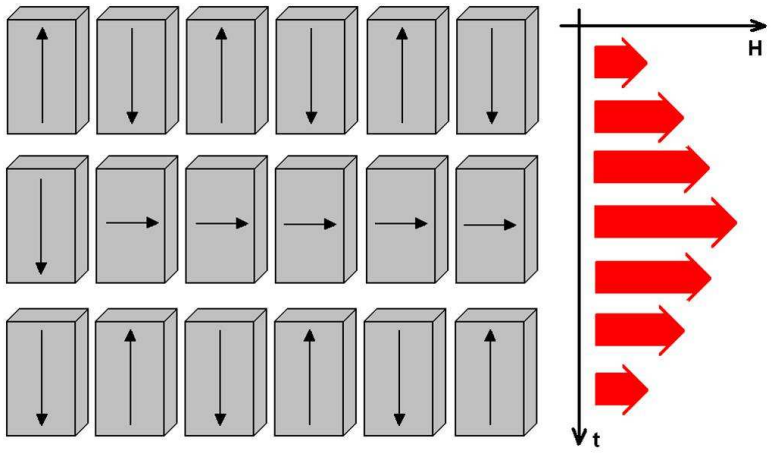



Figure 1.8. NML clock system. Magnets are forced in an intermediate state with an external magnetic field. When the field is removed magnets realign themselves following the input magnet. *M. Vacca et al. "Nanomagnetic Logic Microprocessor: Hierarchical Power Model", IEEE Transaction on VLSI systems, 2012*

#### Clock field generation

As proposed in [2], the magnetic field can be generated through a current flowing through a wire buried under the magnets plane (Figure 1.9). Wires are made of copper and are surrounded by a ferrite yoke to confine the magnetic flux lines [22]. Copper wires height must be accurately tailored, in order to trade off between the

correct magnetic field generated to assure reset and the power consumption due to current flowing.

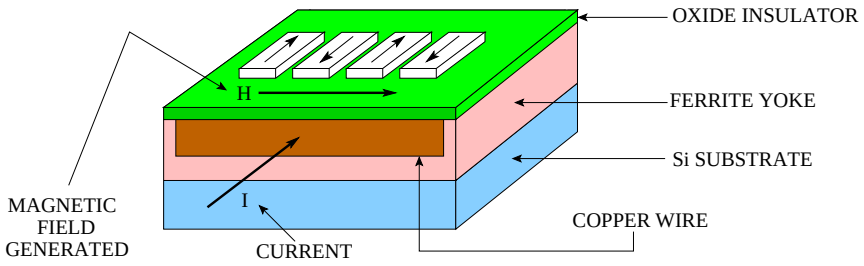



Figure 1.9. Magnetic field generation for MQCA circuits. The magnetic field is generated by a current which flows through a wire placed under the magnets plane. *M. Vacca et al. "Nanomagnetic Logic Microprocessor: Hierarchical Power Model", IEEE Transaction on VLSI systems, 2012*

### 1.2.3 NML logic subtypes

Different clock systems for NML logic were developed, leading therefore to different subtypes of magnetic circuits. This subtypes are summarized in Figure 1.10.

- **In-plane NML (iNML).** This was the first solution ever developed [2], where rectangular shaped ferromagnetic dots are used as basic cell (Figure 1.10.A). Clock is generated by a current flowing through a wire placed under the magnets plane. While many circuits and also the clock system were theoretically and experimentally demonstrated, the losses of the clock system are extremely high. Moreover a precise control on the magnets shape is required. Their main advantage is that they do not require complex magnetic materials. Therefore, coupled with the clock solution presented in chapter 8, they represent the best NML technology available.

- **Multilayered NML (M-NML).** In this case [23] the basic cell is a multilayered structure, a Magneto Tunnel Junction (MTJ), which is made by an insulating layer sandwiched between two layers of magnetic materials (Figure 1.10.B). One layer is made by an hard magnetic material, therefore its state cannot be changed, the other layer is made by a soft magnetic material, so its state can be changed as in iNML circuits. Clock in this case is made using a current flowing through the MTJ itself, which forces the MTJ in the reset state. Compared with iNML this solution offers lower power consumption and a better control thanks to the current based clocking mechanism. They are also the ideal candidate to develop input/output interfaces.

- **Multiferroic NML.** In this solution [24] magnets are made by a thick layer (40nm) of piezoelectric material and a thin (10nm) layer of magnetic material (Figure 1.10.C). Clock should be provided by an externally applied electric field. While theoretically this solution offers an extremely low power consumption and a relatively high speed, the magnets fabrication can be problematic. Magnets aspect ratio is very low (99x101$nm^2$), this not only requires a lithography process with a resolution impossible to obtain, but also the presence of unavoidable process variations will lead to not working circuits. This solution presents also other problems, like difficulties in how to apply the electric field.

- **Out-of-plane NML (oNML).** In this solution [25] dots are made by many layers of Cobalt and Platinum, an the magnetization lies perpendicular to the plane (Figure 1.10.D) thanks to magnetocrystalline anisotropy. Clock is generated by a globally applied oscillating magnetic field perpendicular to the plane. This solution offers lower power consumption with respect to iNML and mostly important it allows to build extremely robust circuits, since magnets can assume any shape. However the clock mechanism is always based on a magnetic field so the power consumption can be still high.

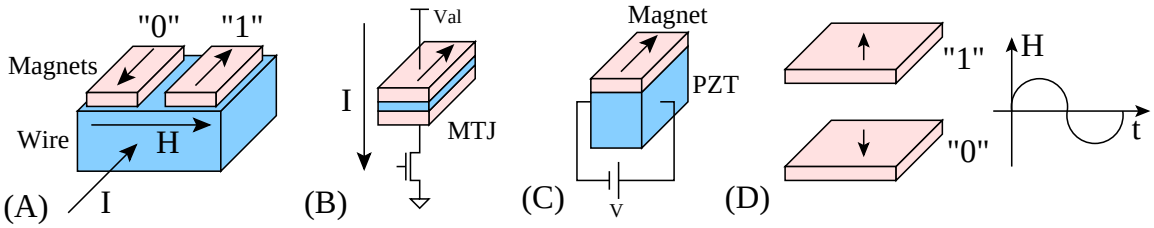



Figure 1.10. NML logic subtypes. A) In-plane NML (iNML) with current generated magnetic field. B) Multilayered NML (M-NML) based on Magneto Tunnel Junctions (MTJ) as basic element. Clock is based on a current flowing through the wire. C) Multiferroic NML. Magnets are multilayered structures made with a layer of piezoelectric material and a layer of ferromagnetic material. Clock should be theoretically generated by an applied electric field. D) Out-of-plane NML (oNML). Magnets are multilayered structures made by Cobalt and Platinum, while clock is an external oscillating magnetic field.

### 1.2.4 3-phase overlapped Snake clock

To propagate the information through the circuit a multiphase clock system is required [26]. In the classic clock scheme phases are 4, however basing on [27], [28] and [5] it is possible to use only three phases, but clock signals must be overlapped.

This solution allows a more easy experimental fabrication with respect to other solutions previously proposed for the multiple-phase clock distribution. It should be noted that this multiple-phase distribution is crucial to guarantee the information propagation without errors in complex nanomagnets arrays. In Fig. 1.11(B), the RESET, SWITCH, and HOLD sequence is shown both in the time and the space axes. In Fig. 1.11(A), the behavior of nanomagnets grouped in the correspondent clock zones is shown. Each clock phase should serve a group of nanomagnets and not a single element. This is due to the unavoidable size difference between the magnets and the metal line that generates the clock signal. When a cell group is in the HOLD phase, the magnets are in the stable "up" and "down" states that store the digital information. These magnets behave like an in- put for the neighbor group which is in the SWITCH state. This means that the previous state of these switching magnets has been already "canceled" due to a reset, and now they are ready to be influenced again. The group in the following region is itself in the RESET state.

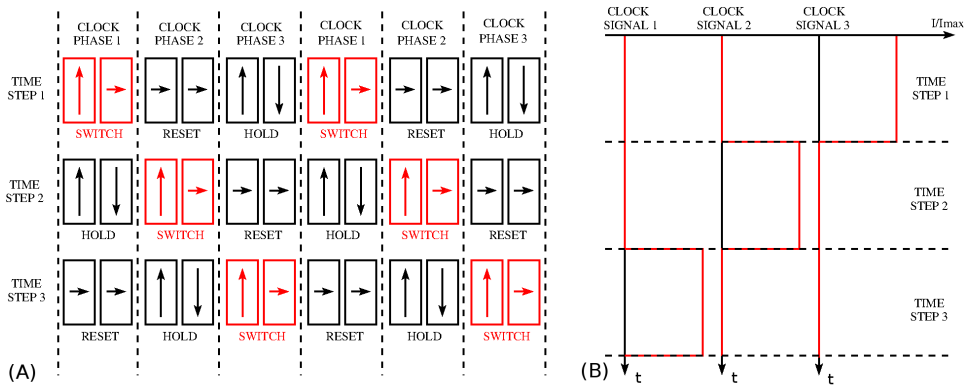



Figure 1.11. (A) Left: logic organization of nanomagnets in time and space following the clock signal sequence (Reset, Switch, and Hold). (B) Right: clock signal on three phases delivered to three different zones in space and repeated in time following the Reset, Switch, and Hold sequence.

### Snake clock layout

The basic idea behind the "snake-clock" system is shown in Figures 1.12 and 1.13. Figure 1.12 shows the layout (Figure 1.12.A) and the 3-D structure (Figure 1.12.B). The nanomagnet arrays can be sandwiched between two thin oxide layers. Metal wires, for clock generation are placed over and under the magnets plane [2]. One wire (phase 1) is straight, while the other wires (phases 2 and 3) are routed in a zigzag style. Wires 2 and 3 are therefore twisted, but, since they belong to two

different planes, there is no interference between them. In this case, for example, phase 2 is routed in the same plane with phase 1, while phase 3 belongs to the bottom plane. Clearly nanomagnets cannot be placed in the area corresponding to the wire twisting, because they will be subject to both phase 2 and 3 clock field. Moreover, due to the wires orientation the direction of the generated magnetic field will have an additional 45 degrees inclination, so that it will not be perfectly parallel to magnets short side, forcing therefore them in the wrong state.

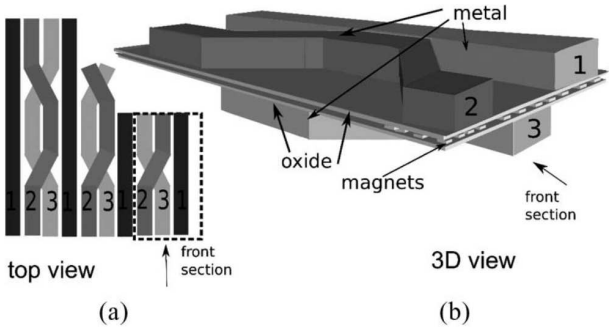



Figure 1.12. Snake-clock. (A) Top view. (B) 3-D view. The 3-D view front section corresponds to the 2-D detail evidenced by the dotted rectangle. Phase 1 is delivered through a straight line on upper plane. Phases 2 and 3 are twisted, but are routed on different planes: phase 2 is on the same plane of phase 1; phase 3 is below the lower plane. Nanomagnets are visible in the section between the two planes. Magnets cannot be placed where wires 2 and 3 are twisted. *M.Graziano, M.Vacca et al."An NCL-HDL Snake-Clock-Based Magnetic QCA Architecture", IEEE Transaction on Nanotechnology, 2011*

Figure 1.13 shows a possible circuit layout based on the "snake-clock" system. The 3D view of the clock zones is sketched without the areas where phases 2 and 3 are crossed, as in those points magnets are not present. It is possible to observe, from Figure 1.13 how the information flow through the circuit. The most important fact is that this clock system allows the signals propagation in every direction, has happens in the classic 4-phase clock scheme. Since to correctly propagate the information through the circuit the correct phase sequence (1,2,3 in Figure 1.13) must be guaranteed; therefore, only a "snake" like propagation is possible. The name "snake" derives from the fact that to propagate signals in "up" or "down" directions signals are routed with a zig-zag style, like the movement of a snake. This is a limitation because it makes the circuit layout quite complex, however the advantage is that this structure is feasible with technology processes currently available, differently from previously proposed solutions.

To assure that the information propagates in the correct direction the three clock signals must be overlapped. When a clock zone in the SWITCH phase, one neighbor

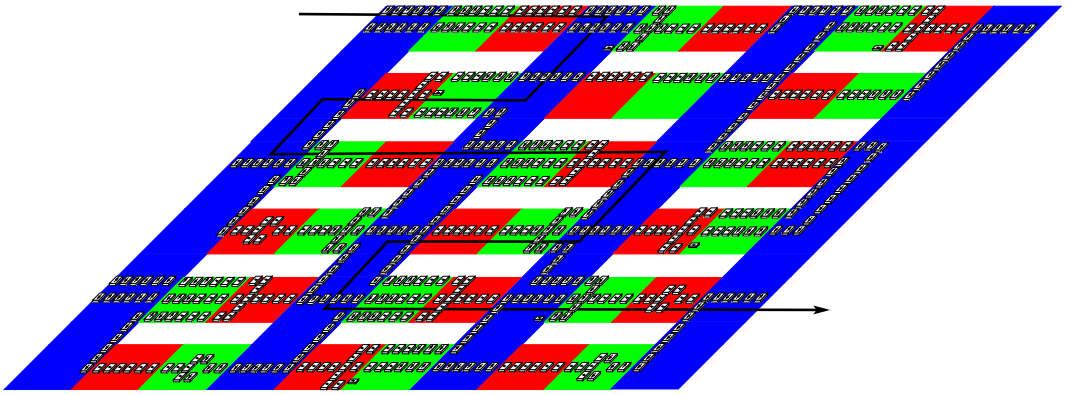


Figure 1.13. An example of circuit based on the "snake-clock" scheme. Different colors of rectangles refer to different clock zone. In white zones no magnets are present because that is the region where two wires are twisted, according to layout in Figure 1.12. *M.Vacca et al."Asynchronous Solutions for Nanomagnetic Logic Circuits", ACM Journal on Emerging Technologies in Computing Systems, 2011*

zone is in the HOLD phase and acts like an input. The other clock zone must be in the RESET phase to not influence the switching zone. However if the magnetic field from the SWITCH zone is removed in the same instant when it is applied to the RESET zone, due to the finite time required to the magnets to switch this will not happens. There will be an instant in which the magnets of the RESET zone are still in the HOLD state, therefore there will be a backward propagation of the signal. To avoid this, clock phases must be overlapped, so that magnets of the RESET zone are forced in the RESET state before the complete removal of the magnetic field from magnets in the SWITCH zone.

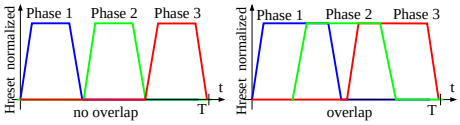



Figure 1.14. Reset field showing a realistic slope. (a) Non-overlapping phases. (b) Overlapping phases, preferred for a correct information propagation. *M.Graziano, M.Vacca et al."An NCL-HDL Snake-Clock-Based Magnetic QCA Architecture", IEEE Transaction on Nanotechnology, 2011*

Fig. 1.15 shows the information propagation through three-phase zones in a sequence of five conditions (snapshots of a continuous time-varying simulation). The reset is applied in sequence on zone 1, then 2, and later 3 with overlap according to Fig. 1.14(b). Basically, to assure the correct information propagation, before cutting off the reset field from a zone [e.g., zone 1 in Fig. 1.15(a)] it is necessary

to apply it to the magnets of the neighbor zone [e.g., zone 2 in Fig. 1.15(b)]. In this way, once the magnets in the previous zone are free from reset [e.g., zone 1 in Fig. 1.15(c)], they can be influenced by the input, as, for example, other magnets in a hold state on the left. This happens without the interference of dots in the following phase [e.g., zone 2 in Figure 1.15(c)]. If this is not done and the reset field is shifted from zone 1 to zone 2 without overlapping, the magnets in zone 2 could still have a vertical magnetization. As a consequence, they could influence backward the magnets in the switching state [29]. A sequence similar to the one just commented allows the information propagation from zone 2 to zone 3.

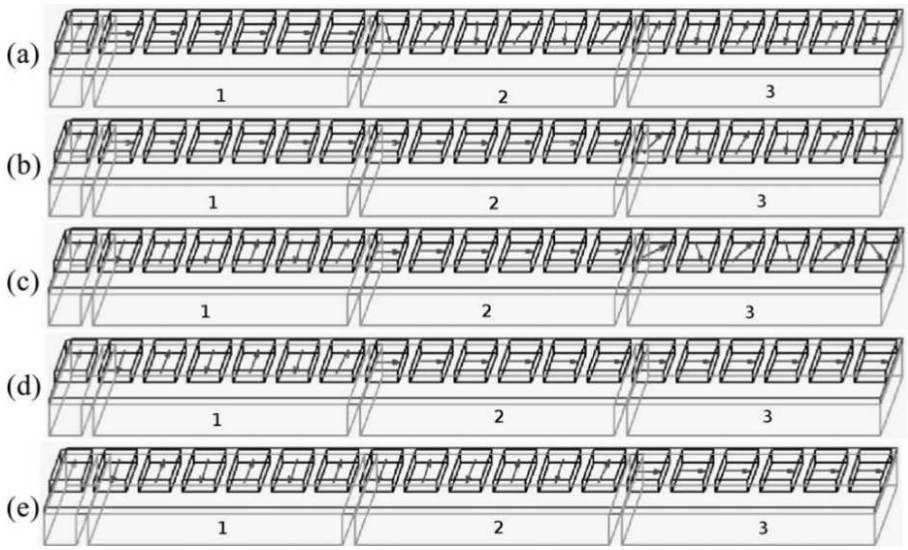



Figure 1.15. Nanomagnet wire information propagation: three phases partially over- lapped. (a) Reset on first zone. (b) Reset on first and second zones. (c) Reset on second zone. (d) Reset on second and third zones. (e) Reset on third zone. *M.Graziano, M.Vacca et al."An NCL-HDL Snake-Clock-Based Magnetic QCA Architecture", IEEE Transaction on Nanotechnology, 2011*

### 1.2.5 Border crosstalk

Figure 1.16.G shows a simulation of the magnetic field generated by one of the clock wires obtained through a Comsol Multiphysics simulation. Thanks to the ferrite cladding around clock wires the confinement of the magnetic field is quite good but not perfect. Magnets placed near the border of neighbor clock zones can feel the influence of the magnetic field, but this is not a problem since a 3-phase overlapped clock system is immune to border crosstalk. The only real constraint of this clock system is that no magnets must be placed where the wires are twisted.

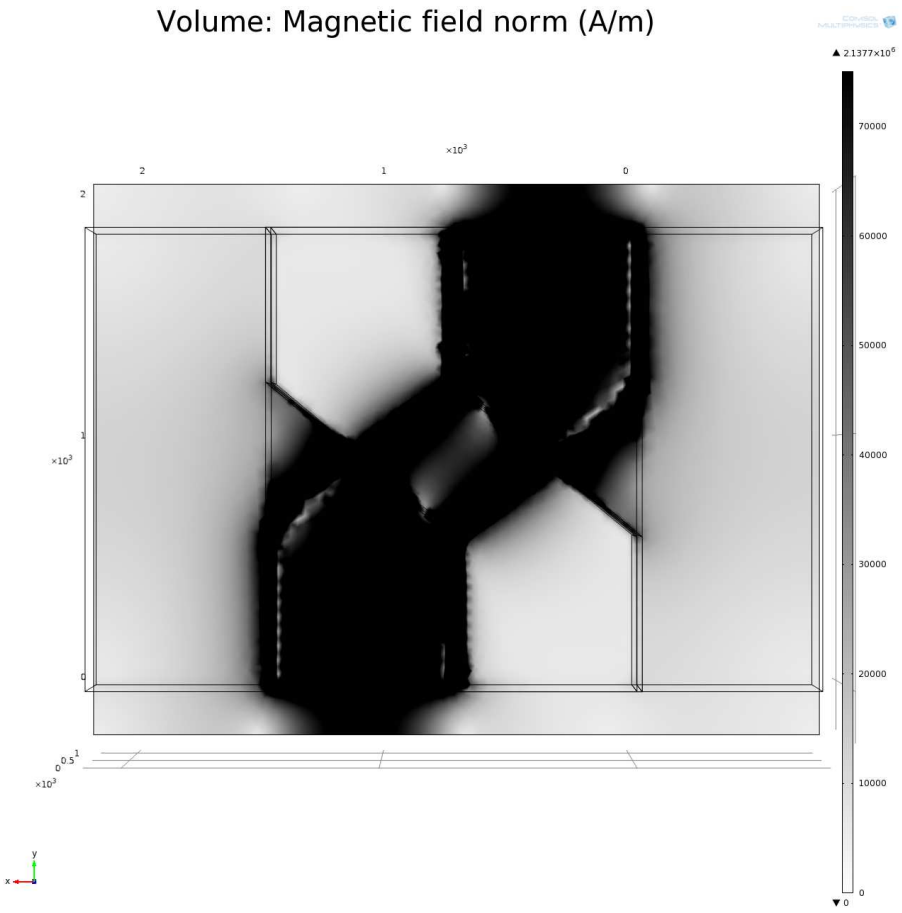



Figure 1.16. Comsol simulation of the twisted clock wires for the Snake-clock scheme. *M.Vacca et al."Nanomagnetic Logic Microprocessor: Hierarchical Power Model", IEEE Transaction on VLSI systems, 2012*

### Snake clock immunity to border crosstalk

The clock mechanism in QCA technology requires a perfect local control on clock zones, as a consequence bad magnetic field confinement can be a problem. Fortunately the 3-phase overlapped clock scheme is immune to the border crosstalk phenomenon. This happens because, when the magnetic field is applied at a clock zone, the zone on their left is in the SWITCH state. If some of the switching magnets are influenced by the neighbor magnetic field they will not switch until the RESET field is removed. As a consequence signal propagation will simply start N magnets before the beginning of the clock zone, where N is the number of magnets influenced by the neighbor magnetic field, but it will propagate correctly. Considering now the clock zone on the right of the zone where the magnetic field is applied,

due to bad confinement some of its magnets can be forced in the RESET state. This is not a problem because clock signals are overlapped so there will be a moment where both clock zones will be in any case in the RESET state.

## 1.3 Two phases clock

While the snake clock system is more realistic than other approaches it has some drawbacks. There is a lot of wasted area due to wire twisting. Moreover it introduces many limitations to the routing of signals and this can lead to an inefficient utilization of circuit area. Recently a new clock system where proposed [6]. It has only two clock phases without signal overlap. The correct propagation direction is assured changing the shape of the element at the beginning of the clock zone. Signals can propagate from left to right or from right to left depending where the special shape magnets is placed. This system simplify the clock generation structure and at the same time it allows signals propagation in each direction greatly reducing the wasted circuit area.

## 1.4 Problems

The presence of a multiphase clock in QCA technology leads to an intrinsically pipelined behavior. To better understand this, a comparison can be done with CMOS circuits. The behavior of a clock zone is equivalent to a register with applied a clock signal similar to the waveform presented in Figure 1.11.B. Clock signals are applied to their correspondent clock zones. Taking as reference Figure 1.13, the first clock signal is applied to all zones of the first color, the second clock signal is applied to all zones of the second color and the third clock signal is applied to all zones correspondent to the third color. The consequence is that every group of three consecutive clock zones has a total delay of one clock cycle. It must be underlined that this is an intrinsic behavior of NML (and more generally QCA) technology, and cannot be changed. The level of pipelining does not depends on circuit layout but on technological constraints.

### 1.4.1 Layout=Timing

The intrinsically pipelined nature of QCA technology generates two important problems. The first one is explained in Figure 1.17.A and 1.17.B, where a MV is reached by three inputs according to two different organizations. The delay of a signal, in terms of clock cycles, depends on the number of clock zone it crosses. Signals must arrive at the inputs of every logic gate (i.e. the MV in this case) at the same time. Therefore the number of clock zones crossed by each input signal must be the same

(Figure 1.17.B). If this does not happen (Figure 1.17.A) the operation result is not correct, because data arrive at different clock cycles. In the simple example shown here it is easy to synchronize signals by controlling the routing. However, in complex circuits only automatic tools could help, but still it may happen that constraints could not be completely satisfied.

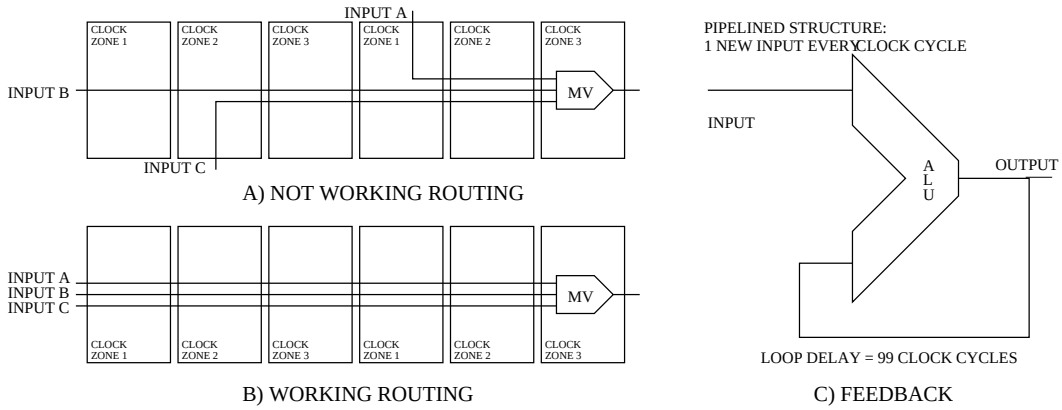



Figure 1.17. QCA problems related to their intrinsic pipelined nature. A) and B) represent the problem of signal synchronization at layout level. A) Shows a case where the circuit will not work properly because the input wires pass through a different number of clock zones. B) Shows a working case with input signals correctly synchronized. C) Schematic representation of the problem of feedback signals. *M. Vacca et al. "Asynchronous Solutions for Nanomagnetic Logic Circuits", ACM Journal on Emerging Technologies in Computing Systems, 2011*

### 1.4.2 Feedback signals

The second problem arises in presence of feedback signals. An example is presented in Figure 1.17.C, where an ALU executes the addition between one input and its own output. Thanks to the pipelined structure, the ALU input arrives at every new clock cycle, but the second input, the feedback, arrives later (in this example after 100 clock cycles) due to the length and therefore the delay of the NML wire. Therefore at every time step the ALU performs the addition between the input and its output result obtained 99 clock cycles before. Changing the length of the input wire does not solve the problem because it simply changes the circuit latency. The circuit will work only if the input is delayed to match the length of the loop. For example, if a new input is sent exactly every 100 clock cycles, and in the meanwhile its value is kept constant, the circuit is synchronized and works correctly. But, if the input arrives with a bigger delay (e.g. 300 clock cycles), the circuit will not work again. This happens because the feedback signal still arrives at the ALU input

after 100 clock cycles. As an example let's suppose the output is 0. The value of the first input (for example '1') is kept constant, therefore the ALU executes the addition between 0 and 1 and gives as a result '1'. At 300 clock cycles the situation is repeated but at this time the two input values are 1 (kept constant) and 1 (the output of the previous operation). This operation gives 2 as a result. At 300 clock cycles a new input is sent, but the output of the ALU will show the wrong value 2 instead of the expected 0. So the circuit works only if the input is delayed of exactly 100 clock cycles. Feedbacks arise therefore serious synchronization problems, better described in 5. Solutions to solve these synchronization problems are also presented in Chapter 5.

## 1.5 NCL logic

A possible solution to automatically solve all the synchronization problems in complex QCA circuits is the adoption of asynchronous logic, like Null Convention Logic$^{TM}$ (NCL, [30]). NCL was proposed in [31] as an ideal candidate for QCA technology thanks to its delay-insensitive nature. In this logic every signal is coded using two bits, which can be in two different states: NULL state when all signals are '0', and DATA state which represents the logic value ('01' means logic '0' and '10' means logic '1'). Signal encoding is shown in Table 1.1.

| X0 X1 | STATO |
|---|---|
| 0 0 | NULL |
| 1 0 | DATA - 0 logico |
| 0 1 | DATA - 1 logico |
| 1 1 | Not admitted |

Table 1.1. NCL dual-rail coding

The delay insensitivity is obtained because circuits switch from NULL to DATA only when all the inputs change from NULL to DATA and they maintains their status until at least one input is in the DATA state. After the completion of the NULL-DATA cycle, before a new data can be accepted from a logic gate, every input must reach the NULL state, completing therefore the DATA-NULL cycle. This assures the circuit operations also in presence of a considerable difference in the propagation delay among the inputs, because gates switches only when all inputs switch. As a consequence adopting this logic solution will solve all synchronization problems of QCA technology. More details are in Chapter 3.

NCL gates are made using an internal loop. For each gate a SET and a RESET equation is defined. The SET equation must be satisfied to switch the gate from

| PORTA | SET | RESET |
|---|---|---|
| TH12 | A + B | A + B |
| TH22 | AB | A + B |
| TH13 | A + B + C | A + B + C |
| TH23 | AB + BC + AC | A + B + C |
| TH33 | ABC | A + B + C |
| TH23w2 | A + BC | A + B + C |
| TH33w2 | AB + AC | A + B + C |
| TH14 | A + B + C + D | A + B + C + D |
| TH24 | AB + AC + AD + BC + BD + CD | A + B + C + D |
| TH34 | ABC + ABD + ACD + BCD | A + B + C + D |
| TH44 | ABCD | A + B + C + D |
| TH24w2 | A + BC + BD + CD | A + B + C + D |
| TH34w2 | AB + AC + AD + BCD | A + B + C + D |
| TH44w2 | ABC + ABD + ACD | A + B + C + D |
| TH34w3 | A + BCD | A + B + C + D |
| TH44w3 | AB + AC + AD | A + B + C + D |
| TH24w22 | A + B + CD | A + B + C + D |
| TH34w22 | AB + AC + AD + BC + BD | A + B + C + D |
| TH44w22 | AB + ACD + BCD | A + B + C + D |
| TH54w22 | ABC + ABD | A + B + C + D |
| TH34w32 | A + BC + BD | A + B + C + D |
| TH54w32 | AB + ACD | A + B + C + D |
| TH44w322 | AB + AC + AD + BC | A + B + C + D |
| TH54w322 | AB + AC + BCD | A + B + C + D |
| THxor0 | AB + CD | A + B + C + D |
| THand0 | AB + BC + AD | A + B + C + D |
| TH24comp | AC + BC + AD + BD | A + B + C + D |

Table 1.2. NCL logic gates.

NULL to DATA and can be seen as the "logic" equation of the gate. The RESET equation must be satisfied to switch the gate from DATA to NULL and it is always the same for each gate. The complete set of NCL logic gates is shown in Table 1.2.

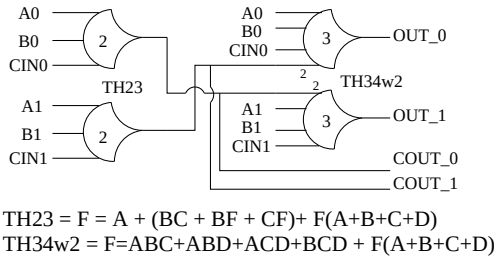



Figure 1.18. NCL circuit example: full adder. Every signal is coded using two bits. Logic gates are TH23 (symbol 2) and TH34w2 (symbol 3) *M.Graziano, M.Vacca et al."Magnetic QCA Design: Modeling, Simulation and Circuits", Cellular Automata Innovative Modelling For Science And Engineering, Intechweb.org, 2011*

Figure 1.18 shows the implementation of a full adder in NCL logic: Two different NCL gates (called TH23 and TH34w2) with their relative encoded signals are used. It is possible to observe that the circuit is split into two specular parts, each of them calculating one of two encoded output bits. This solution has been adapted to general QCA in [31].

## 1.6 Integrated design methodology for nanotechnologies

To keep into account the different nature of NML logic, the fact that it is an emerging technology which characteristics are not fully investigated and to solve the problems that arise from its intrinsic pipelined nature, a new working methodology must be followed. It is based on the idea that device level and architectural level cannot be studied separately. When an architectural solution is studied it must keep into account inputs from the device level research. At the same time research at the device level must keep into account the impact that it has on the circuit architecture. This is quite different from CMOS technology where research at device and architectural level are substantially independent.

The proposed methodology can be summarized according to the flow in Fig. 1.19. It is organized in four steps, each requiring a validation phase. As a result, the design phase may require variations not only to the decisions related to the present step, but also to previous ones. In STEP1, the technology implementation scenario is identified: in our case, the "snake clock". STEP2 entails the study of the proper

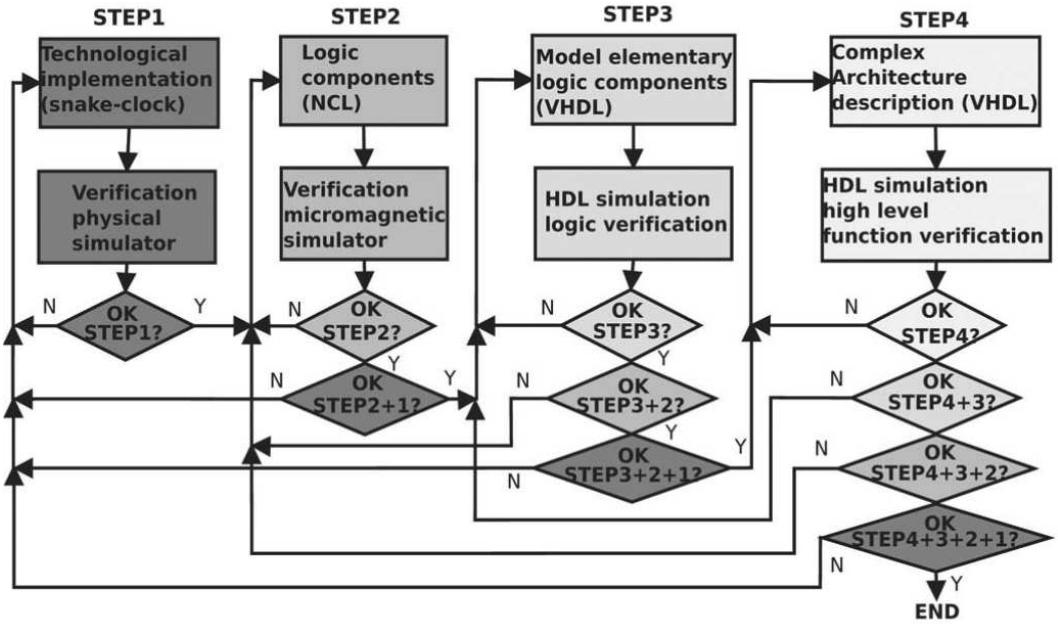



Figure 1.19. Flow diagram of the proposed methodology organized in four steps: (1) technological implementation, (2) logic components definition, (3) HDL model of logic components, (4) architectural HDL description. Each step requires a validation through a proper simulator. Progress from one step to the next is subject to this validation and may require a feedback not only to decision on current step, but on previous ones as well. *M.Graziano, M.Vacca et al."An NCL-HDL Snake-Clock-Based Magnetic QCA Architecture", IEEE Transaction on Nanotechnology, 2011*

logic components that can be adapted to the STEP1 choices: in our case, the NCL gates combined with the "snake-clock" organization. In STEP3, the elementary logic blocks are modeled, taking into account the results from previous steps. Finally, STEP4 consists in designing a complex architecture using the incremental validation results matured up to this point.

# Part I

# Architecture Analysis

# Chapter 2

# NML VHDL modeling

## 2.1 VHDL behavioral model

To describe NML circuits it is possible to build a VHDL model, as was preliminary done in [32][33]. The main idea is to build a CMOS circuits which behaves exactly like its NML counterpart. To obtain this results it must be considerd that, in NML, the behavior of a clock zone is equal to the behavior of a CMOS register: at each clock cicle a new data is sampled. Using therefore registers to model clock zones it is possible to simulate the propagation delay of signals while using ideal logic gates it is possible to emulate the logic behavior of the circuit. In Figure 2.1.A he model of the NCL THxor0 gate is shown while its NML circuit is shown in Figure 2.1.C. Each register is driven by the correspondent clock phase signal, shown in the Figure 2.1.A bottom-right detail.

The resulting simulation of the gate is shown in Figure 2.1.B. The timing behavior of this structure is therefore connected to the clock waveforms applied to each register. The duration of the clock cycle depends on technological constraints, and not on the logic function. The clock period depends in fact on many factors, like the maximum number of magnets in a clock zone, the type of multiphase clock used, if circuit must follow or not an adiabatic switching. All these quantities depend on technology choices.

Figure 2.1.B shows how the output changes from 0 to 1 only when the input A and the input B change from 0 to 1, according to the gate equation. Since there are three clock zones from the input to the output, this means that the circuit has a latency of 1 clock cycle. Output changes from 1 to 0 only when all inputs switch from 1 to 0, otherwise its output remains stable. At this point the cycle can restart.

If a different clock system must be simulated, for example using a 2-phase or a 4-phase clock, it is only necessary to change the number of clock signals applied to the registers and their waveform. As a consequence every clock scheme can be easily

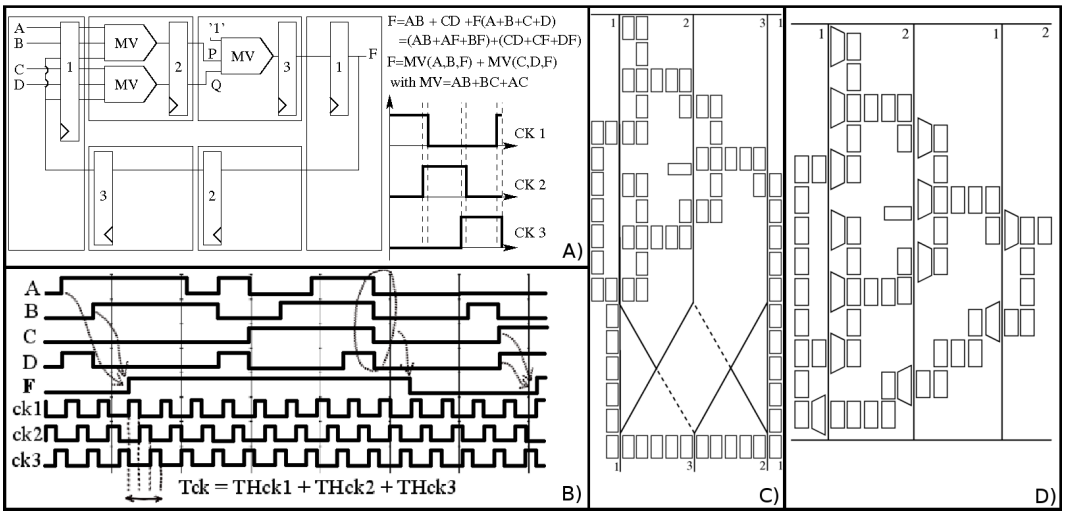



Figure 2.1. A) THxor0 VHDL behavioral model. Logic functions of the gate and the majority voter (MV) are shown in the upper-right detail while the bottom-right detail shows the clock signals applied to each register. B) THxor0 simulation results. It is possible to observe the transition of the gate from F=0 to F=1 when the logic equation is satisfied. C) THxor0 3-phases NML implementation. D) THxor0 2-phases implementation. *M.Vacca et al."Nanomagnetic Logic Microprocessor: Hierarchical Power Model", IEEE Transaction on VLSI systems, 2012*

simulated with this VHDL model. An example is shown in Figure 2.1.D, where a 2-phase clock is used. The model is similar to the circuit of Figure 2.1.C but two clock signals are used instead of three.

## 2.2 Power modeling

There are two main components of losses in NML circuits: The power necessary to force magnets in the RESET state and the losses in the clock system generation, like Joule losses. In Section 2.2.1 for both contributions the dissipation cause, the design parameters upon which the dissipation depends, the model proposed and its VHDL description are described. To generate the clock signals a simple circuit is required [2]. This circuit is made by a limited number of transistors, so it is not considered in the power evaluation because its contribution is negligible.

### 2.2.1 Power consumption components

Figure 1.6.B shows the typical behavior of a nanomagnets during the switching: They follow an hysteresis cycle with an area proportional to the energy spent during the switching. The clock system supply this energy to the magnets when they are forced in the RESET state. When magnets reach again a stable state this energy is normally dissipated in form of heat. Like shown in [34] if the magnetic field is slowly applied to the circuit (adiabatic switching) this power contribution is reduced to $30K_BT$, which represents the average power consumption due to magnet switching. Clearly this power consumption contribution is directly proportional to the number of magnets. The *number of nanomagnets* related to each clock phase is different, because it depends on the layout and circuit complexity. the consequence is that the power consumption in each clock phase is different, however in one entire clock cycle all magnets in the circuit switch. As a consequence, to obtain the total power consumption due to magnets switching during one clock cycle, the total *number of nanomagnets* must be multiplied for the value of $Energy\_mag = 30K_BT$.

The second and most relevant contribution is the power dissipated by *clock wires*, which can be separated in the power stored in the wires inductance and the power lost due to joule effect. Since the clock frequency is relatively low (100 MHz) the energy stored in the inductance is quite low, however if molecular QCA (1 THz) or other NML types like [24] and [23] which works at higher frequencies are considered, this contribution can become relevant. The energy stored in the inductance is therefore considered because this model can be potentially used to simulate Molecular QCA or others NML types. The power dissipated by Joule effect represents the main contribution, mainly because a high value of current is necessary to generate a magnetic field strong enough to force a reset. in both cases power consumption depends on the *length of the wire*, which is a function of the circuit area. This means that it is affected by the circuit complexity and its layout.

### 2.2.2 Model

Directly on indirectly all power consumption contributions in NML logic depend on the number of magnets that compose the circuit. In this model the number of magnets is estimated taking into account the circuit complexity, allowing therefore to estimate the power consumption without knowing the exact layout of the circuit.

Five important points constitute the base of this model (Figure 2.2):

- **Embedded**: The model is embedded in the architecture description: For each blocks there is a part which model the circuit and a part which evaluate the power consumption.
- **Hierarchical**: A block of level $N$ generates information to be propagated to

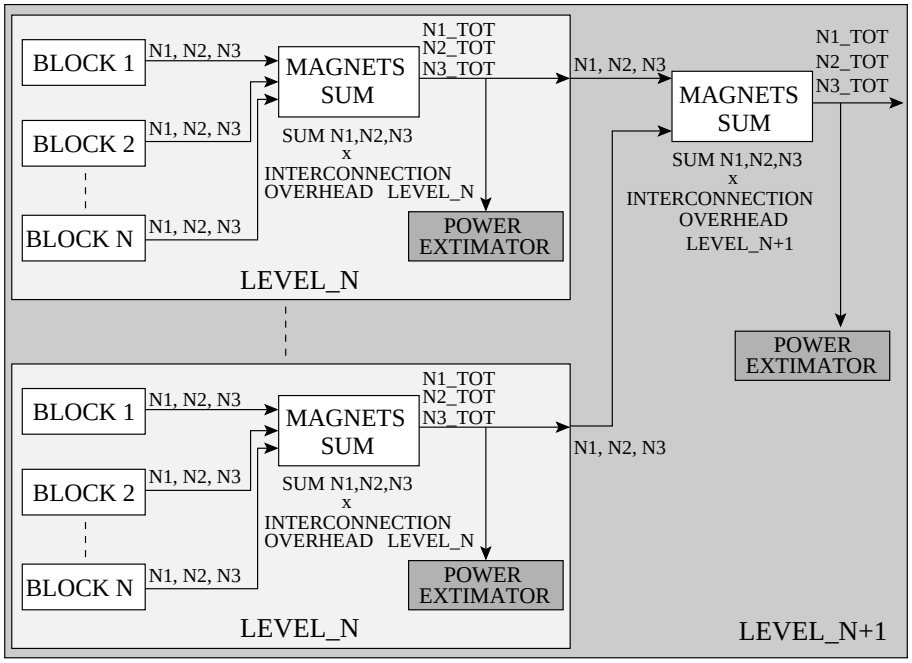



Figure 2.2. Model for the estimation of nanomagnets number in a NML circuit and for the evaluation of power dissipation due to nanomagnets and clock wires. N1, N2, N3 represents the number of magnets (for each clock zone) of each lower level logic block. N1_TOT, N2_TOT, N3_TOT are instead the total number of magnets of the logic level considered. *M. Vacca et al. "Nanomagnetic Logic Microprocessor: Hierarchical Power Model", IEEE Transaction on VLSI systems, 2012*

the higher *N+1* hierarchical level using data on the number of magnets from sub-blocks of level *N-1*.

- **Power estimator**: power consumption for current block $i$ is evaluated by a *power estimator* as a function of the number of nanomagnets in block $i$.
- **Nanomagnet sum**. Inside a logic block a specific block, the *nanomagnet sum*, evaluates the total number of magnets for each clock zone.
- **Overhead**: An *overhead* factor is used to take into account the routing complexity; if sub-blocks have a total sum of magnets equal to $M$, the connection among them could require an additional number of magnets; this *overhead* is estimated and is multiplied by $M$. The *overhead* is different for every logic level.

To obtain the maximum flexibility the use of many configuration parameters is coupled with the power of VHDL language. In the following the model will be

detailed by including portions of VHDL code of the Full Adder (Figure 1.18). Every internal block of level *N* has three output signals, for a specific logic level *N+1* (see Figure 2.2). This signals represent the total number of nanomagnets of that block separated for each phase. Real numbers are used to indicate the total number of magnets to avoid type conversion inside the code. For example, if the block of level *N+1* is the Full Adder, one of the sub-blocks is the *TH23* NCL gate, which *entity* declaration is in the following. The three output port of type "real" can be observed.

```
entity th23 is
  port (a, b, c : in  std_logic;
        y : out std_logic;
        ck1, ck2, ck3 : in  std_logic;
        N1, N2, N3 : out real := 0.0);
end th23;
```

* *M.Vacca et al."Nanomagnetic Logic Microprocessor: Hierarchical Power Model", IEEE Transaction on VLSI systems, 2012*

The logic gate (Figure 1.18) has three input signals *a, b, c* and one output signal *y*. To model the circuit delay three clock signals are used internally for the registers, as mentioned in section 2.1 (Figure 2.1.A). The total number of magnets clocked by a specific phase is indicated by *N1* for phase 1, *N2* for phase 2 and *N3* for phase 3. The TH34w2 gates used in the full adder have a similar entity declaration so they are not reported for simplicity. To model interconnections delay lines (shift registers) are used.

A block, called *MAGNETS SUM*, accepts as inputs the output values holding the number of nanomagnets for each gate instances. This block calculates the sum of all the nanomagnets of the sub-blocks, phase by phase, multiplying them for a factor *interc_overhead* that is used to model interconnections overhead. This factors are located in a package file, which contains all configuration parameters. The *interc_overhead* factor is different for each hierarchical level, because at each logic level interconnections have a different impact on the circuit. The new values are finally used as output of this *SUM* block, which VHDL code is in the following.

```
entity magnets_sum is
  generic(interc_overhead: real := 1.0);
  port(ck: in std_logic;
       f1, f2, f3: in  real_vector;
       N1, N2, N3: out real := 0.0);
end magnets_sum;
architecture behavioural of magnets_sum is
begin

 compute: process (f1, f2, f3)
  variable f1_int: real_vector(f1'Length-1 downto 0)
                                          := (others => 0.0);
  variable f2_int: real_vector(f2'Length-1 downto 0)
                                          := (others => 0.0);
  variable f3_int: real_vector(f3'Length-1 downto 0)
                                          := (others => 0.0);
  variable sum_f1, sum_f2, sum_f3: real := 0.0;
  variable sum_tot_f1, sum_tot_f2, sum_tot_f3: real := 0.0;
```

```
 begin
  f1_int := f1;
  f2_int := f2;
  f3_int := f3;
  sum_f1 := 0.0;

   for i in 0 to f1'Length-1 loop
    sum_f1 := sum_f1 + f1_int(i);
   end loop;
  sum_tot_f1 := sum_f1 * INTERC_OVERHEAD;
  N1 <= sum_tot_f1;

   for i in 0 to f2'Length-1 loop
    sum_f2 := sum_f2 + f2_int(i);
   end loop;
  sum_tot_f2 := sum_f2 * INTERC_OVERHEAD;
  N2 <= sum_tot_f2;

   for i in 0 to f3'Length-1 loop
    sum_f3 := sum_f3 + f3_int(i);
   end loop;
  sum_tot_f3 := sum_f3 * INTERC_OVERHEAD;
  N3 <= sum_tot_f3;

 end process;
end behavioural;
```

* *M.Vacca et al."Nanomagnetic Logic Microprocessor: Hierarchical Power Model", IEEE Transaction on VLSI systems, 2012*

The output signals of the *SUM* block indicate the estimated total number of magnets of the current block. For example in this case they indicate the total number of magnets divided by clock zones for the Full Adder. This values are also exported outside the logic block, and they can be used as inputs by a higher level components but they are also used as inputs for a further element, the *Power Estimator*, which locally finds the power dissipation of this block of level *N+1*. Every logic block, at any hierarchical level, has inside it a *Power Estimator*. This means that, during the circuit simulation, it can calculate not only the total power of the entire circuit, but also the contribution of every component. In the following the VHDL code of the *Power Estimator* is reported. Several parameters are here used: they are defined in a VHDL package and reported in table I and explained in the following description of the model.

```
entity power_extimator is
port(N1, N2, N3: in real;
     ck1, ck2, ck3: in std_logic);
end power;

architecture behavioural of power is
  ... signals definition, initialization....(skipped)
begin
  -- effective circuit area in terms of number of magnets
  Area_eff_1<= N1 * Wasted_Space;
  -- clock wire lenght in terms of number of magnets
  Lwire_1 <= Area_eff_1 / Width_zone_mag;
  -- effective clock wires lenght in meters
  Lwire_eff_1 <= Lwire_1 * (h + vert_space)
```

```
                * h_zone_sep * Wire_curves;
-- effective wire section
Swire       <= (Width_zone - Wire_sep) * Wire_thick;
-- wire resistance
Rwire_1     <= Resistivity * (Lwire_eff_1/Swire);
Log_1        <= (4.0*Lwire_eff_1) / Width_zone;
-- wire inductance
Ind_Wire_1  <= Lwire_eff_1 * 2.0e-7*(LOG(Log_1)-1.0);

Process_power_1: process (ck1) -- PHASE 1
begin
 if ck1 = '1' then
  -- clock power losses due to joule effect
  P_joule_1 <= Rwire_1 * I_max * I_max;
  -- clock power losses due to inductance charging
  P_ind_1 <= ((Ind_Wire_1*I_max*I_max)/2.0)/(T_clock/3.0);
  -- power losses due to nanomagnets switching
  P_mag_1   <= Mag_power * N1;
 elsif ck1 = '0' then
  P_ck_RI_1 <= 0.0;
  P_ck_LI_1 <= 0.0;
  P_mag_1   <= 0.0;
 end if;
end process;

-- effective circuit area in terms of number of magnets
Area_eff_2<= N2 * Wasted_Space;
-- clock wire lenght in terms of number of magnets
Lwire_2 <= Area_eff_2 / Width_zone_mag;
-- effective clock wires lenght in meters
Lwire_eff_2 <= Lwire_2 * (h + vert_space)
                * h_zone_sep * Wire_curves;
-- effective wire section
Swire       <= (Width_zone - Wire_sep) * Wire_thick;
-- wire resistance
Rwire_2     <= Resistivity * (Lwire_eff_2/Swire);
Log_2        <= (4.0*Lwire_eff_2) / Width_zone;
-- wire inductance
Ind_Wire_2  <= Lwire_eff_2 * 2.0e-7*(LOG(Log_2)-1.0);

Process_power_2: process (ck2) -- PHASE 1
begin
 if ck2 = '1' then
  -- clock power losses due to joule effect
  P_joule_2 <= Rwire_2 * I_max * I_max;
  -- clock power losses due to inductance charging
  P_ind_2 <= ((Ind_Wire_2*I_max*I_max)/2.0)/(T_clock/3.0);
  -- power losses due to nanomagnets switching
  P_mag_2   <= Mag_power * N2;
 elsif ck2 = '0' then
  P_ck_RI_2 <= 0.0;
  P_ck_LI_2 <= 0.0;
  P_mag_2   <= 0.0;
 end if;
end process;

-- effective circuit area in terms of number of magnets
Area_eff_3<= N3 * Wasted_Space;
-- clock wire lenght in terms of number of magnets
Lwire_3 <= Area_eff_3 / Width_zone_mag;
-- effective clock wires lenght in meters
Lwire_eff_3 <= Lwire_3 * (h + vert_space)
```

```
                * h_zone_sep * Wire_curves;
  -- effective wire section
  Swire       <= (Width_zone - Wire_sep) * Wire_thick;
  -- wire resistance
  Rwire_3     <= Resistivity * (Lwire_eff_3/Swire);
  Log_3        <= (4.0*Lwire_eff_3) / Width_zone;
  -- wire inductance
  Ind_Wire_3  <= Lwire_eff_3 * 2.0e-7*(LOG(Log_3)-1.0);

  Process_power_3: process (ck3) -- PHASE 1
  begin
   if ck3 = '1' then
    -- clock power losses due to joule effect
    P_joule_3 <= Rwire_3 * I_max * I_max;
    -- clock power losses due to inductance charging
    P_ind_3 <= ((Ind_Wire_3*I_max*I_max)/2.0)/(T_clock/3.0);
    -- power losses due to nanomagnets switching
    P_mag_3   <= Mag_power * N3;
   elsif ck3 = '0' then
    P_ck_RI_3 <= 0.0;
    P_ck_LI_3 <= 0.0;
    P_mag_3   <= 0.0;
   end if;
  end process;

  -- FINAL SUM OF ALL THE CONTRIBUTIONS
  P_joule_tot <= P_joule_1 + P_joule_2 + P_joule_3;
  P_ind_tot   <= P_ind_1 + P_ind_2 + P_ind_3;
  P_mag_tot   <= P_mag_1 + P_mag_2 + P_mag_3;

end behavioural;
```

* *M.Vacca et al."Nanomagnetic Logic Microprocessor: Hierarchical Power Model", IEEE Transaction on VLSI systems, 2012*

For each clock phase the calculation of the power consumption is similar. Equation 2.1 reports the calculation of the first power contribution, the average power dissipated by the nanomagnets.

$$Mag\_power = \sum_{i=1,2,3} N_i \cdot \frac{Energy\_mag}{T\_clock} \tag{2.1}$$

This contribution can be easily evaluated multiplying the number of nanomagnets (e.g. N1 for phase 1) for the energy value of *Energy_mag*, defined in the VHDL package (see table I), and dividing it by the clock period. Both the value for every single phase and the total average values are evaluated.

To evaluate the second contribution of power consumption, which are the losses in the clock generation system, the wires resistance and inductance must be estimated. Since the wire width and thickness are technologically fixed quantities, the only other parameter, that depends on the circuit layout and is necessary to evaluate wires resistance and inductance is the wires length *Lwire_eff_i*. The evaluation of the wires length is therefore the central core of the power estimator block. To estimate the wires length it is possible to start from the number of nanomagnets. This can be done because the number of nanomagnets and the area of the circuit

are related. The wire width depends on the width of the clock zone, defined as a parameter in the package (*Width_zone* in table I). In this case, as an example, its value is 700nm that is equivalent to the width of 10 nanomagnets (*Width_zone_mag*). The basic idea behind the estimation of wires length is shown in Figure 2.3, using different parameters to takes into account the wasted area. This clock structure is based on the three phases pattern (one straight and the other two twisted) repeated M times [5], depending on the circuit area and shape.

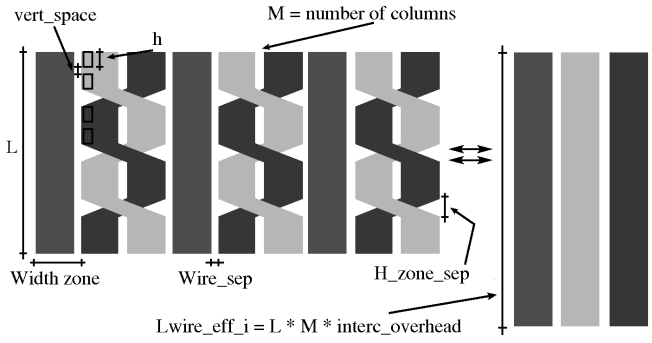



Figure 2.3. Wire length calculation. Wires of same color are connected serially, so they can be approximated as one straight wire. A factor *Wire_curves* is used to takes into account wire angles overhead. *M. Vacca et al. "Nanomagnetic Logic Microprocessor: Hierarchical Power Model", IEEE Transaction on VLSI systems, 2012*

The connections among wires of the same phase (with the same color in Figure 2.3) are serial connections located outside the perimeter of the circuit, where no magnets are present. All the wires that generate the clock field, of a specific phase, are therefore equivalent to a single clock wire with an equivalent length *Lwire_eff_i*, which is approximately M times bigger than the original length $L$ given by one side of the circuit rectangular perimeter. The wire length evaluation starts estimating the total number of magnets, which can be also considered as the area of the circuit expressed in terms of number of magnets. This value is multiplied for a constant (*Wasted_space*) that take into account the separation area among magnets that are part of different logic gate (see equation 2.2 for a single phase). This separation area is equal to the area of one magnet and it is necessary to avoid crosstalk.

$$Area\_eff\_i = Ni \cdot Wasted\_Space \tag{2.2}$$

Since this area is expressed in terms of number of nanomagnets, it can be divided for the width of the clock zone which is also expressed in number of magnets. The results is the length of the clock wire (equation 2.3), expressed in terms of number of magnets.

$$Lwire\_i = \frac{Area\_eff\_i}{Width\_zone\_mag} \tag{2.3}$$

The length of the wire expressed in meters, *Lwire_eff_i*, can be easily obtained multiplying the sum of the physical height of the magnet $h$ and the vertical separation between magnets *vert_space*, for the wire length expressed in terms of magnets previously evaluated. Two other constants are introduced to obtain a better evaluation of the wires length: the height of the vertical separation between clock zones *h_zone_sep* and a factor used to take into account the wires curves *Wire_curves*, the segment of wire used to connect parallel wires of the same clock phase.

$$\begin{aligned} Lwire\_eff\_i = Lwire\_i \cdot (h + vert\_space) \cdot \\ \cdot h\_zone\_sep \cdot Wire\_curves \end{aligned} \tag{2.4}$$

If the value of *h_zone_sep* is set to 1 no vertical separation between clock zones is considered. This is useful to evaluate the length of clock wires in case of a 2-phase clock. The wire section can be calculated

$$Swire = (Width\_zone - Wire\_sep) \cdot Wire\_thick \tag{2.5}$$

as the product between the effective wire width (*Width_zone* - *Wire_sep*) and its thickness (*Wire_thick*). Knowing the wire section and length it is possible to calculate the resistance *Rwire_i* for each phase, which is given by the well known equation 2.6. The metal resistivity is a parameter that can be set accordingly to the material chosen.

$$Rwire\_1 = Resistivity \cdot \frac{Lwire\_eff\_1}{Swire} \tag{2.6}$$

The wire inductance *Ind_Wire_i* calculation is approximated as if the wire were straight and alone. No mutual inductance among neighbor wires is considered. For the purpose of this model this approximation is sufficient, because it gives the order of magnitude of the inductance. It is calculated according to equation
2.7.

$$\begin{aligned} Ind\_Wire\_i = Lwire\_eff\_i \cdot 2e{-}7 \cdot \\ \cdot ln\left(\frac{4{*}Lwire\_eff\_i}{(Width\_zone - Wire\_sep)} - 1\right) \end{aligned} \tag{2.7}$$

Due to the flexibility of this model, if the accuracy of this evaluation is not sufficient, the inductance calculation can be easily improved substituting this equation

with a more precise one. It is now possible to evaluate power dissipated thanks to Joule losses, which can be calculated by the equation 2.8,

$$P\_joule\_tot = \sum_{i=1,2,3} Rwire\_i \cdot I\_max^2 \tag{2.8}$$

where the value of the current $I\_max$ is chosen equal to 1mA [35].
The power dissipated by the inductance is calculated as in equation 2.9,

$$P\_ind\_tot = \sum_{i=1,2,3} \frac{\frac{1}{2} \cdot Ind_{Wire_i} \cdot I\_max^2}{\frac{1}{3} T\_clock} \tag{2.9}$$

where $T\_clock$ is the total clock period defined in the package.

This method for calculating the number of magnets and the power dissipation is repeated at every logic level, where the output signals of the lower level blocks become the input signals of the higher level SUM block. This allows to propagated the number of nanomagnets from the lowest and simple level, the logic gates, to the highest and more complex level. For the elementary logic gates, the Majority Voter and the inverter, the number of nanomagnets is defined as a constant in the package, since it is well known and fixed.

It is important to underline that the effectiveness of this model relies on the value chosen for the constant used to take into account overheads (interconnections overhead, wire curves, ...), and most important, on the level of accuracy used in the circuit description. To obtain the more accurate simulation constants used in this model were extrapolated starting from NML theory and low level simulations, taking into account all the physical and layout constraints actually known. However, since up to now there are no tools for automatic place&route of NML circuit it is impossible to know the exact layout of complex NML circuits. As a consequence data obtained from this model will provide only approximated results. This model gives the best results when different architecture of the same circuit are compared, as shown in Chapter 3. Results of this model applied to a NML microprocessor are shown in Chapter 3.

Table 2.1. Parameters and constant defined in the VHDL package used in the NML power model. *M.Vacca et al."Nanomagnetic Logic Microprocessor: Hierarchical Power Model", IEEE Transaction on VLSI systems, 2012*

| **Parameter name** | **Default value** | **Explanation** |
|---|---|---|
| h_zone | 6.0e-7 | Height of clock zone (meters) |
| h_zone_sep | 2.0 | Vertical separation between zones, where no magnets are allowed. |
| Width_zone | 7.0e-7 | Width of a clock zone (meters) |
| Width_zone_mag | 10.0 | Width of a clock zone in terms of magnets |
| Wire_thick | 6.0e-7 | Clock wire thickness (meters) |
| Wire_sep | 2.0e-8 | Space between clock wires (meters) |
| Wasted_Space | 2.0 | Relative area used by components (magnets + separation spaces) |
| h | 1.0e-7 | Height of a nanomagnet (meters) |
| W | 5.0e-8 | Width of a nanomagnet (meters) |
| vert_space | 2.0e-8 | Vertical separation space between magnets (meters) |
| oriz_space | 2.0e-8 | Horizontal separation space between magnets (meters) |
| Wire_curves | 1.1 | Overhead due to path wires for connecting different wires pieces |
| OV1_logic_gate_level | 1.1 | Interconnect overhead inside a logic gate |
| OV2_intermediate_level | 2.0 | Interc. overhead in small clusters of logic gates (full_adder, mux, registers..) |
| OV3_logic_block_level | 1.5 | Interconn. overhead inside functional element (alu, counter,....) |
| OV4_top_level | 1.2 | Interconnect overhead due to logic blocks interconnection |
| I_max | 1.0e-3 | Maximum current flowing in clock wires (Ampere) |
| clock_overlap | 11.0 | Clock overlap percentage |
| Resistivity | 1.78e-8 | Clock wire resistivity |
| T_clock | 9.0e-9 | Clock period (seconds) |
| Energy_mag | 30.0*KT | Energy associated to a single magnet switch (with T=300 and K=boltzman constant) |
| N_mag_mv | 5.0 | Number of magnets in a Majority Voter |
| N_mag_inv | 7.0 | Number of magnets in an Inverter |

# Chapter 3

# NML architecture level analysis

## 3.1 4 bit microprocessor

To analyze the real potential of NML logic a good benchmark is required. Since microprocessors are one of the most diffused digital electronic circuits today available, and moreover they contain both combinational and sequential circuits, a microprocessor represents a very good benchmark to test the true potential of NML logic. The processor here described is based on [36] but substantially improved and it is implemented using 3 different asynchronous solutions: with full NCL logic, with a mixed NCL-boolean solution and using full boolean logic, adopting an ad-hoc communication protocol the maximizes performances and minimizes the circuit area. The microprocessor is described using the VHDL model described in Chapter 2.

### 3.1.1 Full NCL logic

#### General architecture

Figure 3.1 shows the microprocessor architecture implemented with NCL logic. Four main components compose the microprocessor. A PROGRAM COUNTER which aim is the generation of instruction memory address and is also capable to handle jump instructions. An INSTRUCTION MEMORY is used to store the instructions that must be executed. Since the address uses 4 bits the INSTRUCTION MEMORY has a total of 16 memory cells. A DATA MEMORY with 4 memory cells is used to temporary store operation results. An arithmetic/logic unit (ALU) is the computational core of the microprocessor and is capable of arithmetic operations (addition, subtraction) and logical operations (bit wise AND/OR). The structure of the microprocessor is very simple but it allows the execution of most instructions that are present in modern machines, allowing therefore the validation of NML (and QCA) technology.

NCL logic, like every asynchronous logic, requires a communication protocol to

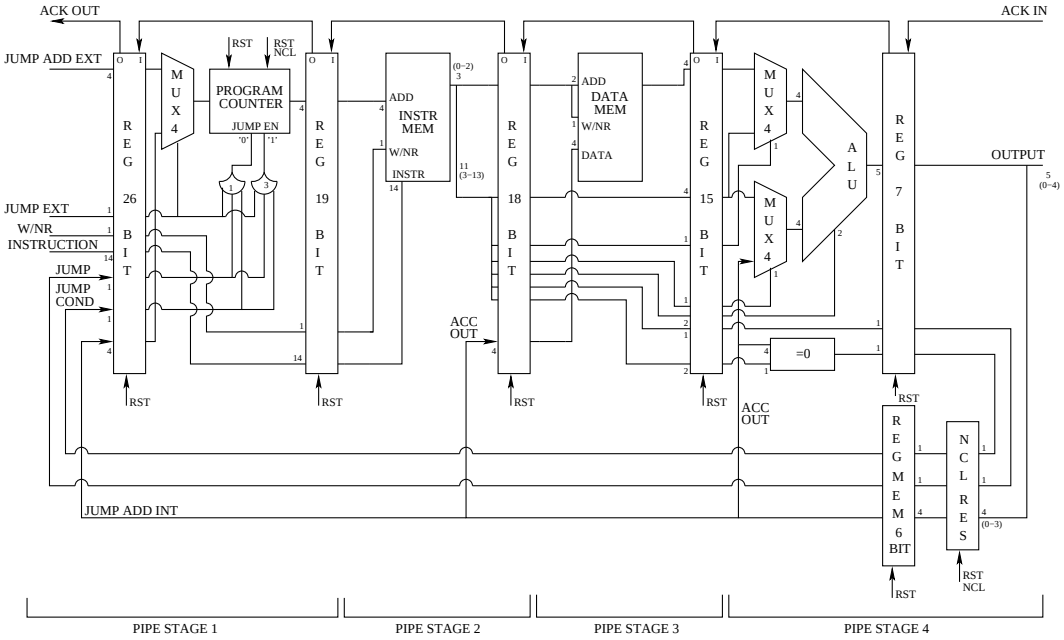



Figure 3.1. NCL Microprocessor architecture.

operate. As a consequence the microprocessor is divided into four stages, separated by asynchronous registers. This can be gathered by figure 3.1, where every block of combinational logic is embraced by two asynchronous registers that generate and exchange this communication protocol:

- A DATA is propagated from a register output to the input of the next one through the combinational circuit.
- At this point the register receives the DATA and sends back an acknowledgement (ACK) to the previous register.
- When the ACK is received at the first register, a NULL (all the outputs to '0') is sent through the combinational circuit.
- The final register receives the NULL and sends back another ACK signal.
- Once this second ACK signal is received the first register is ready to accept a new data.

The first pipeline stage contains the program counter necessary for the address generation. If a jump is required, the two NCL gates TH13 ("1" in symbol and function $F = A + B + C + F(A + B + C)$) and TH33 ("3" in symbol and function

$F = ABC + F(A + B + C)$) generate the signal which forces the output of the program counter to the desired value. The multiplexer selects the jump address from different sources: an address externally generated by other blocks or internally originated by the ALU. The second and third pipe stages contain two memory blocks. The memories organization is based on [37].

The last pipeline stage contains the data path, organized in an Arithmetic Logic Unit (ALU), an accumulator register (ACCUMULATOR) and a zero comparator (=0) to implement conditional branches. Two multiplexers allow to choose, for two ALU inputs, from a combination of three sources: data memory, accumulator or immediate (from the instruction memory). This enables many different arithmetic/logic operations to be executed.

### NCL registers

NCL registers are quite different from their CMOS counterpart. They have no memory ability and their only purpose is to implement the asynchronous communication protocol. The architecture of a generic register is shown in Figure 3.2.

The register is composed by two TH22 NCL gates for each bit (so that for example a 12bit register has 24 TH22 gates). This is due to the dual-rail encoding of NCL logic, where logic is always duplicated. Each TH22 has two inputs: the data signal and the ACK IN signal, so that a new DATA is accepted only when the ack signal is received from the next stage. A majority voter is connected at the beginning of each TH22 gate to force the inputs in the NULL state (all '0'). One of the inputs of the majority voter is fixed to '0' so it works like an AND gate. When the RESET pin is forced to '0' the register goes in the NULL state. This is an asynchronous reset that is used at the beginning to force the circuit in a known state. Each output of the register is connected to a net of NCL gates that generates the ACK OUT signal for the previous stage. When a new DATA is accepted the ACK OUT become '0', indicating that the register is ready to accept a NULL value. When the NULL value is accepted the ACK OUT become '1', indicating that a new DATA can be sent.

### Feedback in NCL

Feedbacks in NCL require some tricks. In order to work a feedback signal must be always in the opposite state of the other parts of the circuit. So when the circuit is in the DATA state the feedback signal must be in the NULL state and when the circuit is in the NULL state the feedback signal must be in the DATA state. To achieve this result a particular structure must be used. This structure, composed by three asynchronous registers, is shown in Figure 3.3.

This 3 asynchronous registers are cascaded and the ACK OUT signal of each

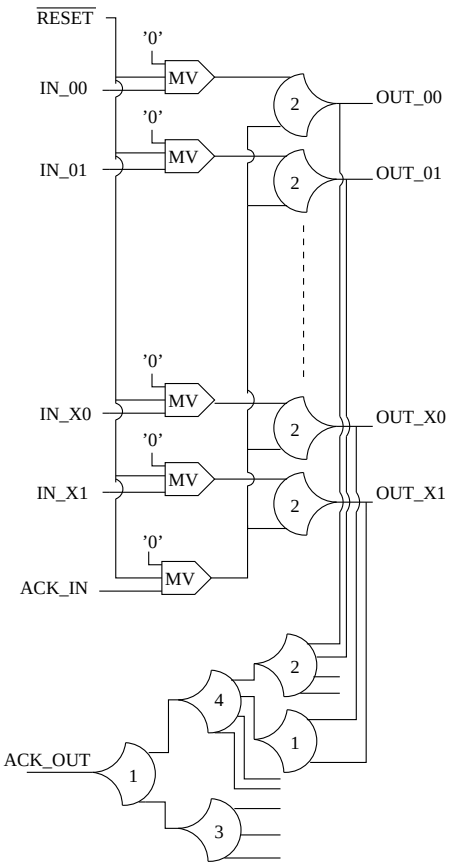



Figure 3.2. Generic asynchronous register architecture.

register is connected to the ACK IN port of the previous one. Due to this configuration the output register is always in the opposite state of the other two registers [27]. As a consequence adding this block in each feedback loop assures the correct operation of the circuit. The delay block between the first and second register is necessary to assure the correct circuit initialization.

### Circuit initialization

To work properly the whole circuit must be correctly initialized in the NULL state. To do so the initialization must proceed in two step: first all the asynchronous register (also the 3 register in each loop) are forced in the NULL state setting the RESET pin to '0'. Then a special block, called "Reset NCL" is used to correctly initialize the 3 loop registers. This block force a DATA in the first of the 3 registers. Due to the internal delay placed between the first and the second register (Figure 3.3) the last register goes in the DATA state. Now that the microprocessor is correctly

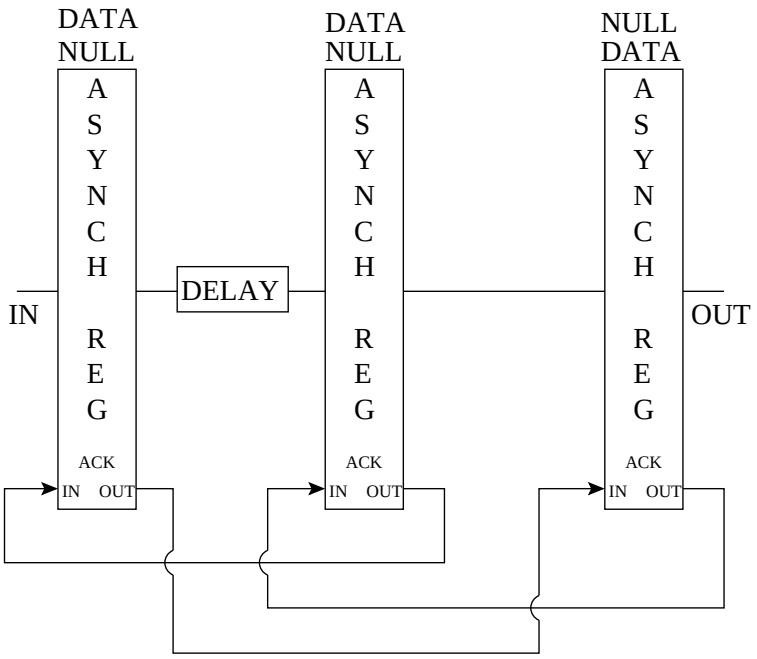



Figure 3.3. NCL feedbacks structure.

initialized instructions can be sent to its inputs.

### Alu

The Alu is based on two different units. An arithmetic unit which is based on a ripple carry adder and is capable of addition and subtraction operations. The reasons behind the choice of a ripple carry adder, which in CMOS is one of the slowest architectures, are explained in 4. In QCA technology, if there are constraints on the clock zones layout, the simplest the architecture is the better the performance are. The second core part is the NCL logic unit, which is very simple and designed to perform AND and OR operations on two operands. Two NCL multiplexers are also present: the first one selects, for the second operand of the adder, between signals $B$ and $\overline{B}$. This action, combined with the possibility to select the carry in, enables subtraction instructions. It is worth to underline that NCL logic does not need inverters, because, due to the particular encoding adopted, to obtain the inverse of a signal wires must be simply switched (in the NCL encoding the inverse of "01" is "10"). The final multiplexer selects between the output of the ripple carry adder and the output of the logic unit, depending on the type of operation chosen. Two logic gates TH12 (symbol "1" and function $F = A + B + F(A + B)$) and TH22 (symbol "2" and function $F = AB + F(A + B)$) assure that the overflow signal

switches to 0 when a logical operation is performed.

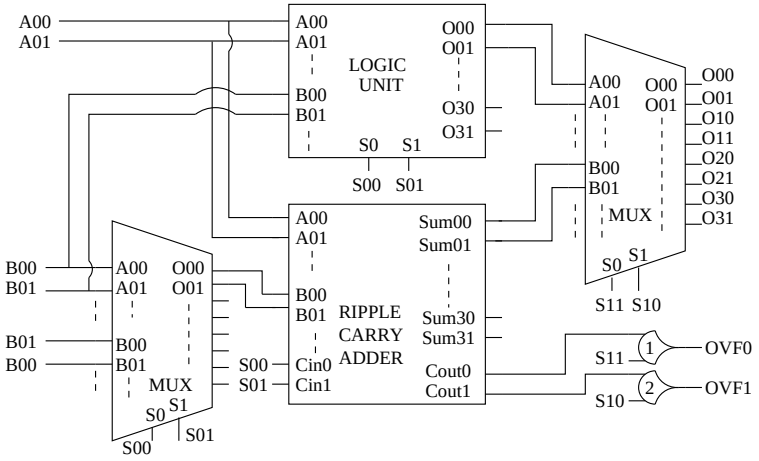



Figure 3.4. NCL magnetic QCA arithmetic/logic unit architecture. *M.Graziano, M.Vacca et al."Magnetic QCA Design: Modeling, Simulation and Circuits", Cellular Automata Innovative Modelling For Science And Engineering, Intechweb.org, 2011*

The architecture for a 1 bit multiplexer is shown in Figure 3.5. It is made by 4 TH54w22 gates and two TH12 gates. To extend this architecture to N bits this circuit must be repeated N times.

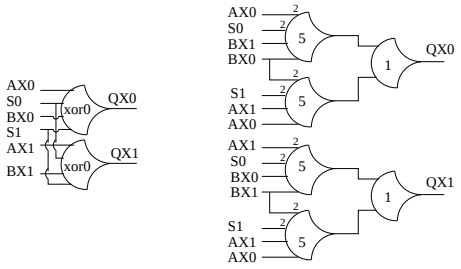



Figure 3.5. NCL mux architecture. *M.Graziano, M.Vacca et al."Magnetic QCA Design: Modeling, Simulation and Circuits", Cellular Automata Innovative Modelling For Science And Engineering, Intechweb.org, 2011*

### Program counter

The *program counter*, shown in the figure 3.6, is built around a ripple carry adder, with one of the inputs fixed to logic '1'. and the other input connected to its own output. As a consequence at every new cycle its state is increased by one, generating

therefore a sequential address for the INSTRUCTION MEMORY. One of the ripple carry adder inputs must be fixed to the logic value '1', to have a unitary increment. Unfortunately NCL circuits must periodically switch from NULL state ("00") to DATA state ("01" or "10") and viceversa. Therefore if one input is kept fixed to "10", the circuit does not work. To overcome this problem a particular block must be used: It generates "10" when the counter is in the DATA state, and "00" when the counter is in the NULL state. A similar block, which generates a fixed logic 0 ("01") is used for the carry in of the ripple carry adder. The "REG MEM" block represents the 3 loop registers required for the internal loop. To implement jump instructions a multiplexer allows to choose as next address a value coming from the outside, transforming therefore the counter in a program counter. A "Reset NCL" block initializes the loop during the reset phase performed when the circuit is booting.

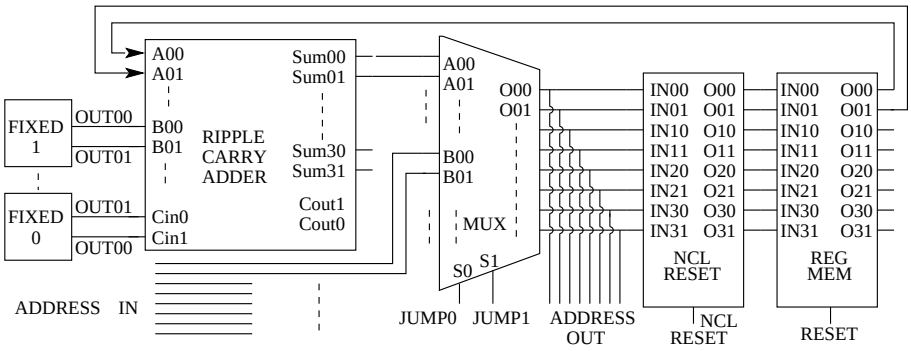



Figure 3.6. NCL magnetic QCA program counter architecture.

### Memory

The memory architecture is shown in Figure 3.7. It is a matrix of N*M cells (16 rows and 14 columns for the instruction memory and 4 rows and 4 columns for the data memory), where the NCL memory cell is shown in the detail. It is very a complex sequential circuit with the 3 loop register and the "Reset ncl" block inside.

A row decoder and an output selector made by many multiplexers are used to select the correct memory row and its correspondent output.

### Instruction set

The microprocessor architecture is very simple, but can execute many type of instructions, like memory read/write, jump and arithmetical/logic ones. The full instruction set is reported in Figure 3.8. The first four bits are used to select from

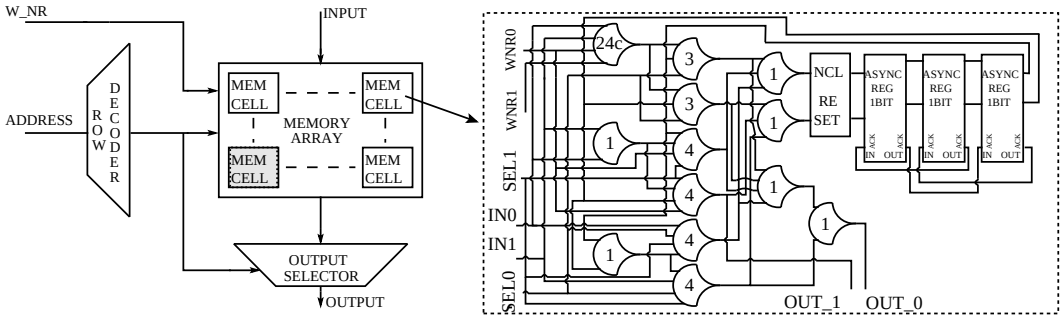



Figure 3.7. NCL parallel memory architecture.

outside a specific address in the instruction memory. This is useful during the microprocessor programming. The fifth bit must be put to '1' (DATA 1, "10" in the NCL case) when is necessary to jump to a specific address of the instruction memory from outside. The sixth bit allows to choose between memory programming and program execution.

The others 14 bits constitute the microprocessor instruction subfield. They are the input of the instruction memory. The first bit is used to choose between write or read in the data memory. The second and third bit indicate the data memory address. Bits from the forth to the seventh are used to indicate an immediate value. The eighth and ninth bit are used to select the inputs for the alu. The tenth and eleventh bit select the alu operation. The twelfth bit is used for unconditional jump, while the last two are used for conditional jumps.

Instructions are divided in four groups: arithmetic/logic, memory read and write, jump, and extra instructions. Arithmetic/logic instructions are four: addition, subtraction, logic OR and logic AND. These operations can be performed on different types of operands, i.e. between the memory and the accumulator, between the memory and the immediate, or between the immediate and the accumulator.

Read and write instructions provide the interface with the data memory, while jump instructions allow the execution of relatively more complex programs. Two are the possible jump instructions: conditional and unconditional. Both of them allow a direct jump to a location provided by the instruction immediate, or an indirect jump to an address stored in memory. However only the unconditional one allows an indirect jump to an address stored in the accumulator register. The conditional branch is based on the zero condition: It happens only if previous ALU result is zero.

Some other instructions are a byproduct of how the architecture was designed. In particular it is possible to execute some instructions while a result is written in the data memory. These operations are the four arithmetic/logic operations, between

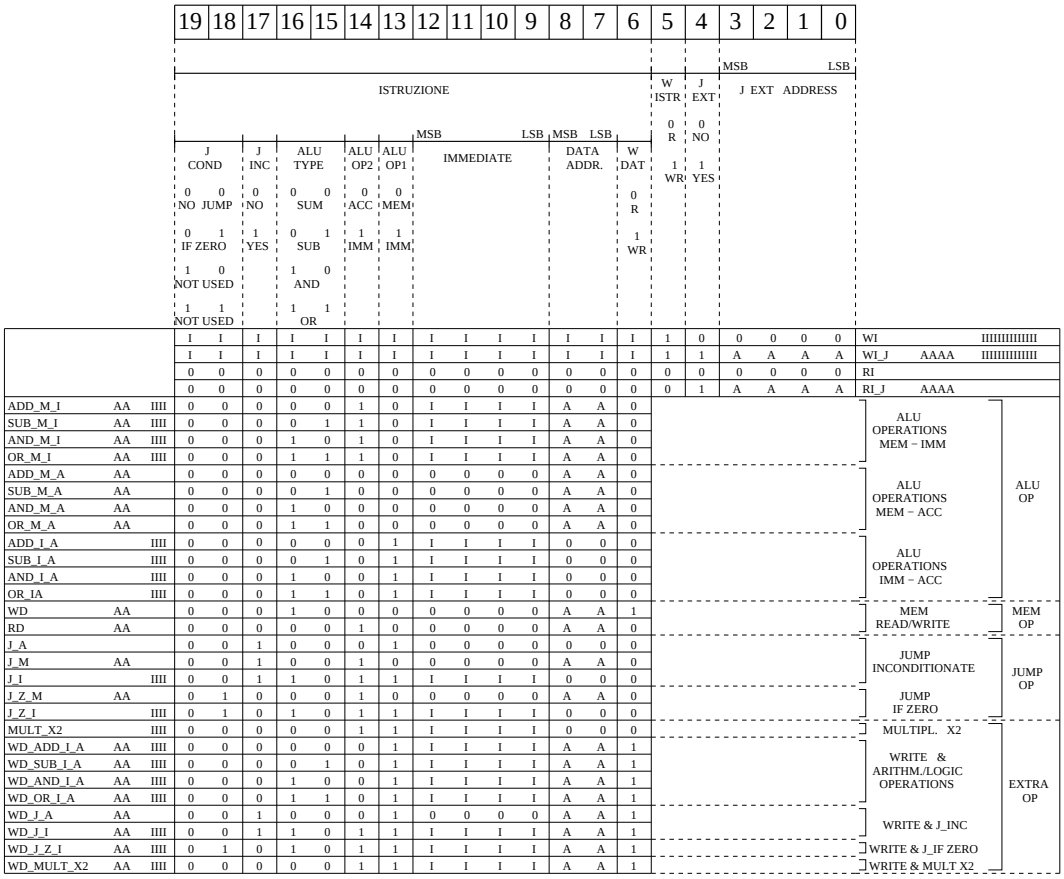



Figure 3.8. Microprocessor instruction set.

the accumulator and the immediate, and some jump instructions. More in detail, it is possible to execute a direct unconditional jump and an indirect one to the address stored in the accumulator, as well as a direct conditional jump. Finally it is possible to perform a multiplication by 2 of an immediate, which is equal to a shift operation, and it can be executed both independently or during a memory writing.

While most of existing microprocessors are built around the instruction set, this case is different. In this microprocessor the instruction set is derived from the architecture. The consequence is that the resulting set of operations is quite different from the standard set of instructions of existing microprocessors. However this is not a problem, because the purpose of this microprocessor is only to test different logic solutions for NML technology and to verify how much general purpose architectures are suited for this technology.

### Microprocessor testing

To verify the microprocessor correct behaviour and performance we have used as benchmarks several simple algorithms and here report a division and a base 2 logarithm. Their code is in figure 3.9.

DIVISION

| | | | |
|---|---|---|---|
| 0 | WD_ADD_I_A | 00 | 1100 |
| 1 | WD | 01 | |
| 2 | SUB_M_I | 01 | 0011 |
| 3 | WD | 01 | |
| 4 | ADD_M_I | 00 | 0001 |
| 5 | WD | 00 | |
| 6 | RD | 01 | |
| 7 | J_Z_I | | 1001 |
| 8 | J_I | | 0010 |
| 9 | RD | 00 | |
| 10 | JD | | 0000 |

LOGARITHM

| | | | |
|---|---|---|---|
| 0 | ADD_I_A | | 1111 |
| 1 | WD | 00 | |
| 2 | SUB_M_I | 00 | 1000 |
| 3 | J_Z_I | | 1100 |
| 4 | SUB_M_I | 00 | 0100 |
| 5 | J_Z_I | | 1101 |
| 6 | SUB_M_I | 00 | 0010 |
| 7 | J_Z_I | 00 | |
| 8 | SUB_M_I | 00 | 0001 |
| 9 | J_Z_I | | 1111 |
| 10 | SUB_M_I | 00 | 0001 |
| 11 | WD_J_I | 00 | 0010 |
| 12 | SUB_I_A | | 1111 |
| 13 | SUB_I_A | | 1111 |
| 14 | SUB_I_A | | 1111 |
| 15 | SUB_I_A | | 1111 |

Figure 3.9. Division and logarithm program code.

The results are approximated because this processor cannot execute floating point operations, unless proper operands are used. Simulations are based on Modelsim [38], and resulting waveforms are shown in figure 3.10 for the division benchmark. Only essential signals are reported for the sake of simplicity.

As described in Chapter 1, NCL signals are coded using two bits. Referring to waveforms in the figure, when the signal marked with (0) has value “1” and the corresponding signal marked with (1) has value “0”, the associated information is a logic zero. In the opposite situation, a logic one is the associated information, and when both signals are “0”, the signal is in the NULL state. To summarize: if (1)(0) = “01” the value is 0, if (1)(0) = “10” the value is 1, if (1)(0) = “00” the

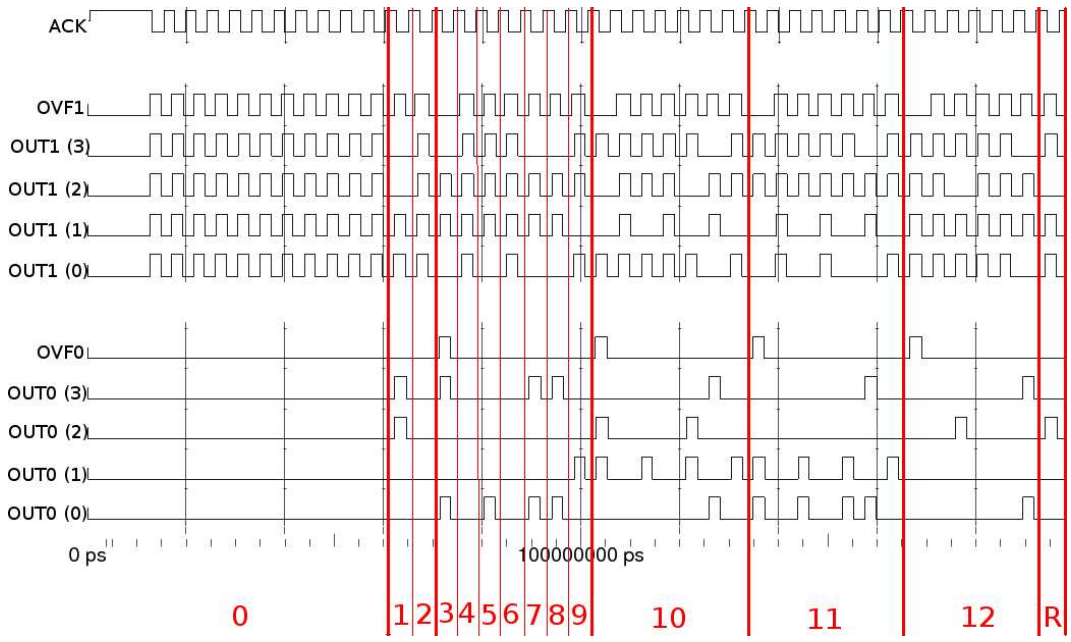



Figure 3.10. Simulation results of the the division algorithm executed on the pure NCL microprocessor. In the mixed case waveforms are identical, but the time of the execution is reduced. *M. Vacca et al. "Asynchronous Solutions for Nanomagnetic Logic Circuits", ACM Journal on Emerging Technologies in Computing Systems, 2011*

value is NULL. Thus all signals switch from DATA to NULL and again to DATA (i.e. from "10" or "01" to "00" and so on). The time reference for the processor, as it is asynchronous, is the previously mentioned ACK signal. Numbers in the figure bottom line refer to sequences of operations commented hereinafter.

During the simulation initial part (numbered 1 in the figure bottom line) the program is loaded in the instruction memory (internal processor signals are not shown for the sake of brevity), therefore output signals are always 0 (i.e. DATA correspond to "01"). In the phase 2 the algorithm execution begins with operands initialization: first by storing value 0 (0000) in the data memory, which is a counter variable, and then by storing value 12 (1100 i.e. "10", "10", "01", "01" for Out4, Out3, Out2, Out1 respectively), which represents the value that must be divided. In phase 4, number 3 (0011), which is the second operand of the division, is subtracted to number 12, thus the output assumes value 9 (1001, i.e. "10", "01", "01", "10"). In phase 5 the subtraction result is stored in the data memory. Output bits in this moment show the value 0000. During phase 6 the counter variable is incremented by 1 (i.e. "01", "01", "01", "10" is shown at the output), and then stored in the data memory (phase 7). In phase 8 the result of the subtraction, previously stored,

is read: if this value is equal to zero the program jumps to the end (phase 9). Differently, if the result of the subtraction is different from 0, the program jumps back to the third instruction and the cycle restarts (phase 10).

Phases from 4 to 10 are repeated for other 3 times (sequences 11, 12, 13) and in each the counter variable is incremented by 1 and the number 3 is subtracted again. Finally, during the phase 13, the subtraction gives a 0, therefore the conditional jump is true and the program jumps to the end. In this situation the counter variable, which represents the result of the division, is read (phase 14). This value is 4 (0100, i.e. '01", "10", "01", "01"), which is the correct result of the operation 12/3.

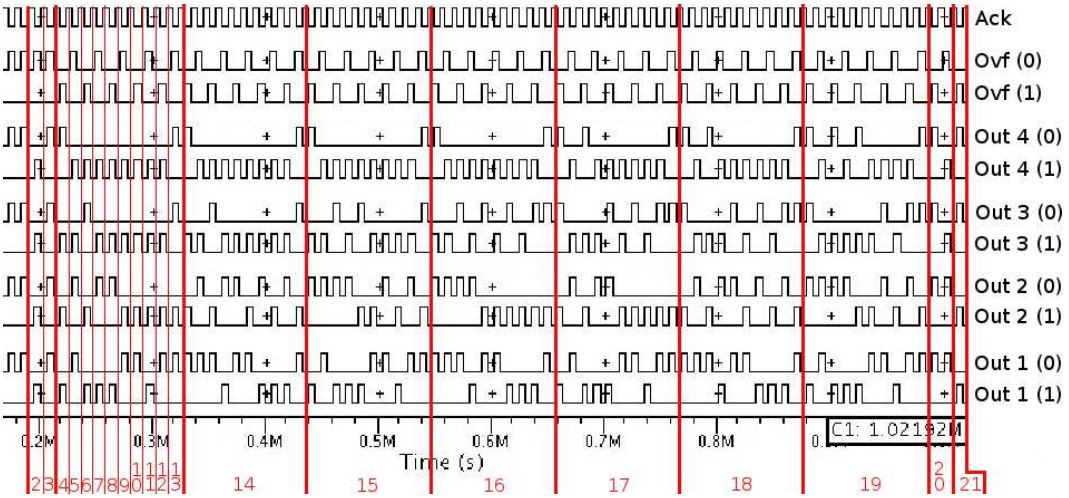



Figure 3.11. Logarithm algorithm simulation results (starts from step 2 for space reason as step 1 concerns just initialization).

Simulation waveforms of the base 2 logarithm are shown in figure 3.11. This algorithm is much longer than the previous one, because it requires more instructions to complete the execution. Also in this case in phase 1 the program loading occurs, therefore microprocessor outputs are 0: this part has been cut from the figure as less meaningful. In phase 2 the program execution starts with an instruction which forces the output of the ALU to value 15 (1111), which is the number chosen in this test for the logarithm calculation. This value is stored in the data memory during phase 3. During phase 4 number 8 (1000) is subtracted from it, and, if the result of this operation is zero the program jumps to the end (phase 5). On the contrary, if it is different from 0, a new cycle begins. Next 6 instructions are quite similar to previous two. The value 1111 is subtracted first with 4 (0100) during the phase 6, with 2 (0010) during the phase 8 and with 1 (0001) during the phase 10. After each of these phases a conditional jump is performed (phases 7, 9 and 11), only if the

result of each subtraction is zero. If, as happens in this case, every subtraction gives results different from 0, during phase 12 the value, on which we want to calculate the logarithm, is decreased by 1 unit, and its value is overwritten in the data memory in phase 13, while an unconditional jump, back to instruction number 2, is performed. This cycle is repeated seven times (sequences 14, 15, 16, 17, 18, 19, 20), and finally, during the last cycle, the result of the first subtraction becomes 0. As a consequence the program execution is interrupted, a jump to the end is executed, and the result of the operation is displayed (phase 21). This value is 3 (0011), which is the correct approximated result of the base 2 logarithm of 15.

### Performances

Microprocessor performance are shown in Table 3.1. The time required for the execution of an instruction is about $5.35\mu s$, which is around 1000 times bigger than the clock period used (5.46ns). This can be easily explained considering how the NCL logic assures the delay insensitivity. The behavior of QCA circuits is pipelined for what concerns magnetic signal propagation, but the asynchronous protocol freezes the circuit from the logic point of view and accepts a new data only after the completion of the DATA-NULL cycle. The propagation time of the signals through the circuit, and the propagation time of the ACK signal, are equal to the latency of the combinational circuit. This means that an asynchronous register accepts a new data only after a time equal to 4 times the circuit latency (one time for the propagation of the DATA, one time for the propagation of the NULL and two times for the propagation of the ACK signals). However, since NML have a pipelined nature, the propagation time of the wires in terms of clock cycles (latency) can be very high, therefore the operations are stopped for a very long period. It is important to underline this point: A pure synchronous Boolean NML circuit has a throughput of 1 data for every clock cycle, due to its pipelined nature, but only combinational dataflow circuits are allowed (no feedbacks). Therefore an hypothetical Boolean NML processor could execute one instruction at every clock cycle, i.e. every $5.46ns$. But since feedback cannot work, this kind of microprocessor cannot be really used in its pure form. The NCL solves the synchronization problems allowing the construction of any kind of circuits at the cost of dramatically decreasing the overall speed.

Table 3.1 shows the estimated total power consumption of the microprocessor, thanks to magnets and clock wires. The evaluation of the power consumption is embedded in the VHDL model and follows the methodology shown in Chapter 2. The total power consumption is $63.8\mu W$ which is a very high value, compared to the results found using the other solutions discussed in Section 3.1.2 and 3.1.3. This is due to the high number of magnets that compose the microprocessor, about 4 millions of nanomagnets. This outline one of the intrinsic characteristics of NML (and QCA) technology. Power consumption and latency depend on the circuit area,

so the bigger the area is the highest the power consumption and latency are. In Chpater 4 this fact is verified comparing the layout of two adders, the Pentium 4 adder and the ripple carry adder, showing that this last one has the smallest area between them.

Table 3.1. NCL Microprocessor performances. *M.Vacca et al."Asynchronous Solutions for Nanomagnetic Logic Circuits", ACM Journal on Emerging Technologies in Computing Systems, 2011*

| | NCL |
|---|---|
| Istruction execution time [$\mu$s] | 5.35 |
| Area (number of nanomagnets) | $4*10^6$ |
| Nanomagnets power dissipation [$\mu$W] | 23.9 |
| Clock power dissipation (Joule Effect) [$\mu$W] | 39.99 |

To summarize the results for this implementation it is possible to say that adopting NCL completely solves the NML synchronization problems. Moreover the circuit fabrication is simpler, because gates can be placed without worrying about signal synchronization. However there is an important drawback: The circuit area is significantly higher, and the area increment generates a proportional increment in power dissipation and a decrement in circuit speed. However the huge area increase depends mainly on memories, therefore, if memories are implemented in boolean logic a huge gain in performance can be expected. Generally speaking this results highlight that NCL logic is not particularly suited for sequential circuits.

### 3.1.2 NCL-Boolean logic

#### Architecture

In NCL logic the most critical parts are the sequential circuits, and memory cells are made using sequential circuits. The area of the memory cells is therefore quite big so memories have a huge impact on circuit area and performance. To enhance performance the idea is therefore to design memories using boolean logic, leading therefore to a mixed NCL/Boolean microprocessor. This solution is based on the assertion that in NML technology combinational circuits have good performance, and are also less complicated to implement from the synthesis point of view. However particular attention must be paid to the synchronization of signals to avoid the layout=timing problem. Therefore, if the combinational parts are implemented using Boolean logic, the performance can be substantially improved, since the number of magnets is reduced. Asynchronous registers are still necessary to synchronize signals propagation, particularly in case of feedback signals.

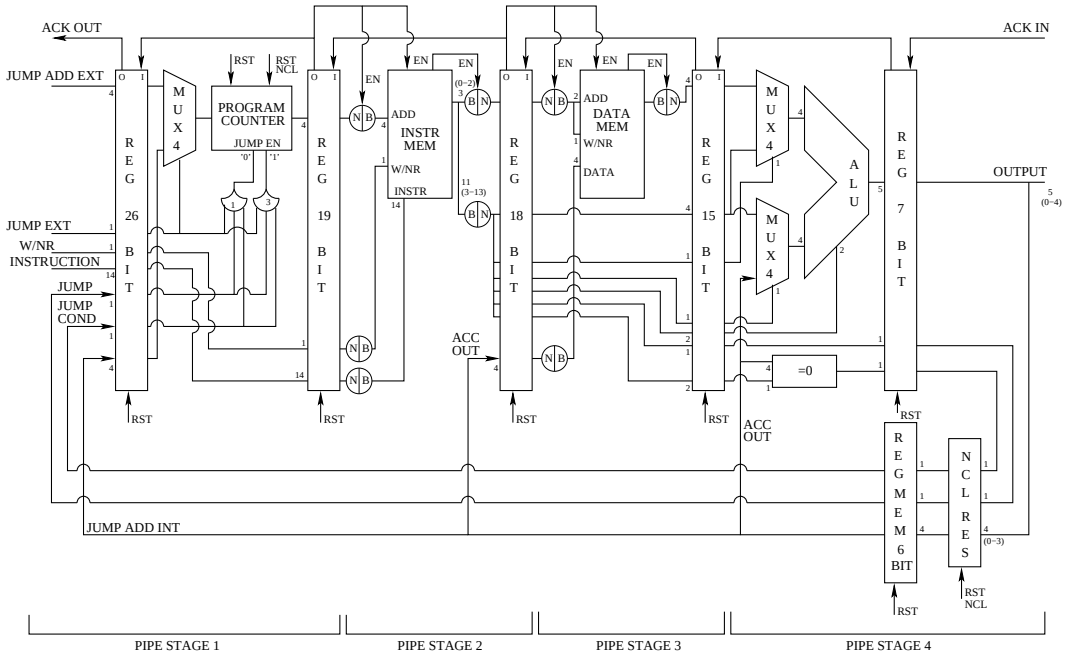



Figure 3.12. Mixed logic microprocessor architecture. Memories are designed using boolean logic, interfaces are therefore required.

The architecture of the mixed NCL/Boolean microprocessor is reported in Figure 3.12. The differences with respect to the pure NCL version consist in the Boolean blocks (the two memories) and in the interfaces (B/N and N/B) blocks in Figure 3.12. Only the two memories are implemented in boolean logic and the other components are left in NCL.

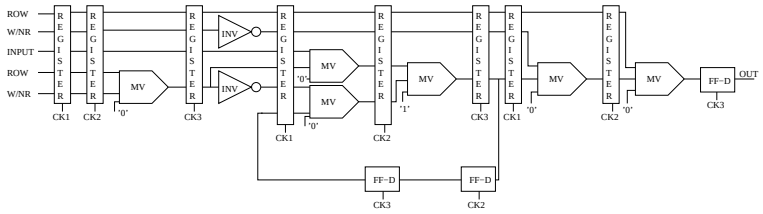



Figure 3.13. Boolean memory cell.

The Boolean memory cell is shown in Figure 3.13. It is simpler than its NCL counterpart, and thanks to its regularity it is more likely to keep under control the delays due to magnetic wires (layout=timing). This would be for sure more complicated in a sparse logic block.

### Interfaces

Two interfaces are necessary to encode/decode signals from Boolean to NCL and from NCL to Boolean. The *Boolean-NCL* logic interface is simple, because it has only to split the Boolean signal in the two bits according to NCL encoding (Figure 3.14).

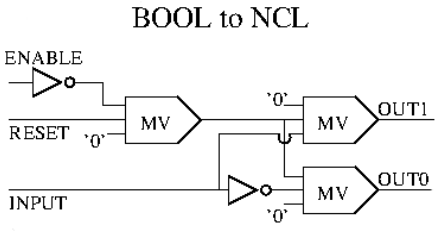



Figure 3.14. Boolean-NCL interface. *M. Vacca et al. "Asynchronous Solutions for Nanomagnetic Logic Circuits", ACM Journal on Emerging Technologies in Computing Systems, 2011*

The *NCL-Boolean* logic interface is more complicated. This is due to the necessity of including a memory loop inside the interfaces (Figure 3.15). NCL switches periodically from NULL to DATA, but Boolean logic is always in the DATA state. As a consequence, this interface not only has to merge the two bits encoding them in one single bit, but also it has to maintain the value stored when the NCL logic is in the NULL state.

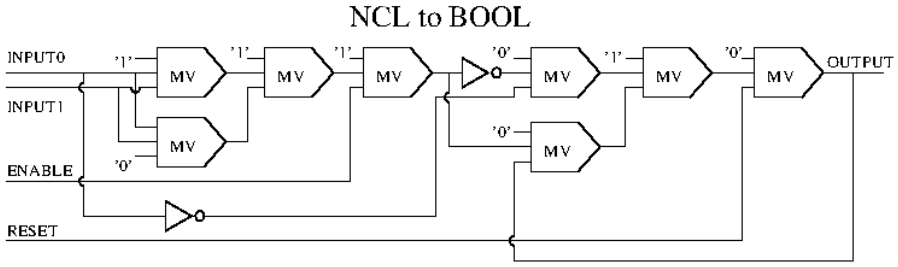



Figure 3.15. NCL-Boolean interface. *M. Vacca et al. "Asynchronous Solutions for Nanomagnetic Logic Circuits", ACM Journal on Emerging Technologies in Computing Systems, 2011*

### Enable signals for interfaces

The two interfaces must guarantee the synchronization between the two logic topologies. Therefore the interfaces use an *ENABLE* signal which arrives from the previous stage. This signal is different from the *ACK* signal, which arrives from the next stage. The enable signal is generated by the logic block placed before the interface, and it is generated only when that block has updated its output. In this way the interface performs the signal conversion only when the signals are already arrived at its inputs, avoiding glitches and false commutations.

### Boolean Memory

The boolean memory has an architecture similar to its NCL counterpart. A matrix of memory cells is used to store the information.

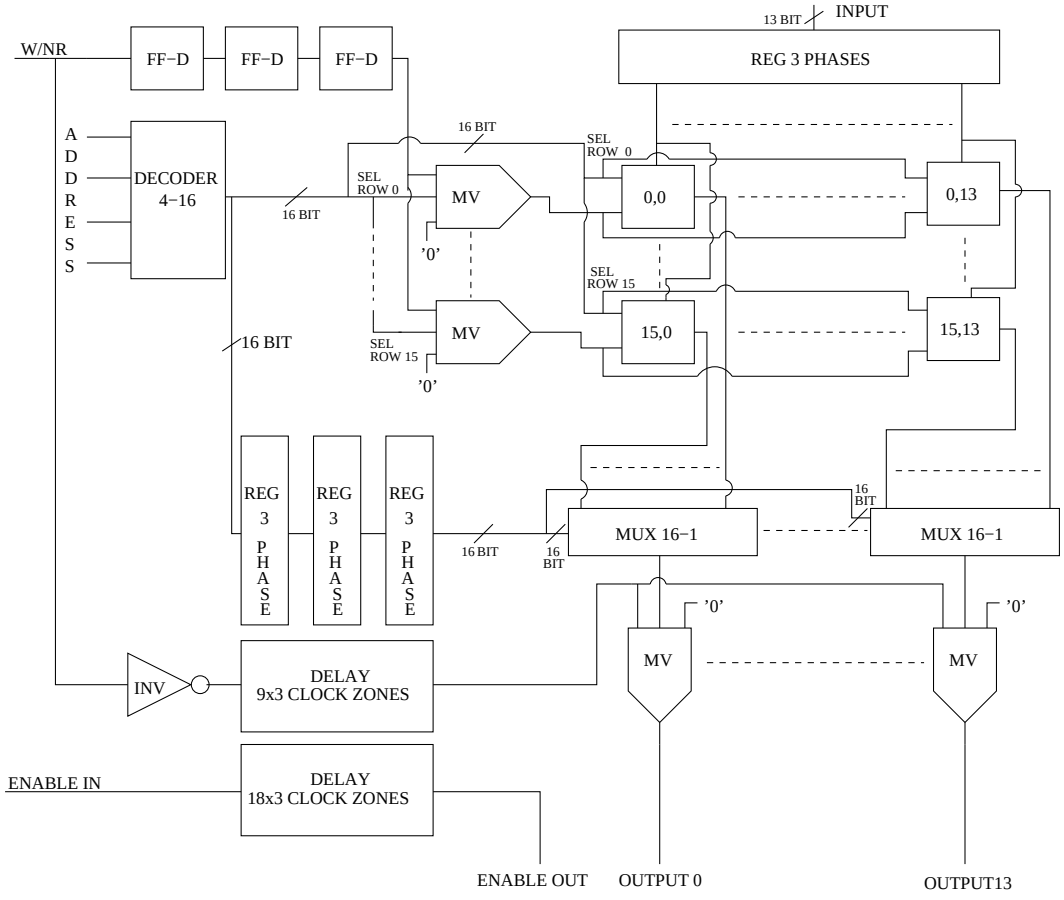



Figure 3.16. Boolean memory architecture.

A decoder is used to select the matrix row, in Figure 3.17 is shown the implementation of a 4 to 16 decoder.

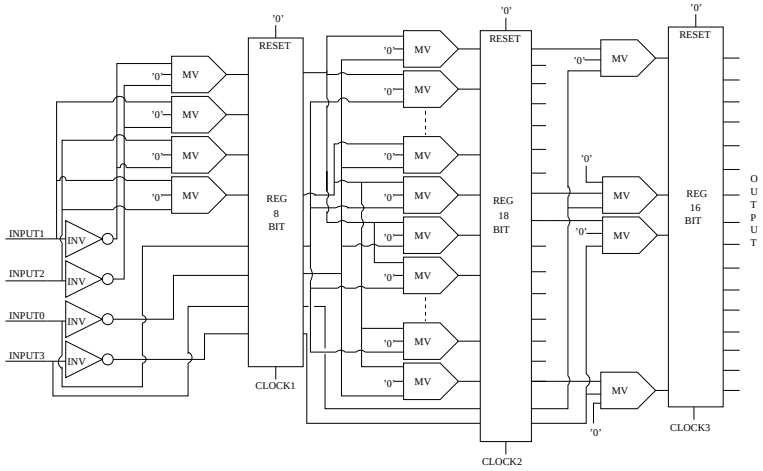



Figure 3.17. A 4 to 16 decoder made using boolean logic.

N Mto1 multiplexers are used to select the correct output, where N is the number of columns of the memory matrix and M is the number of row. Multiplexer selection bits are driven by the the decoder output.

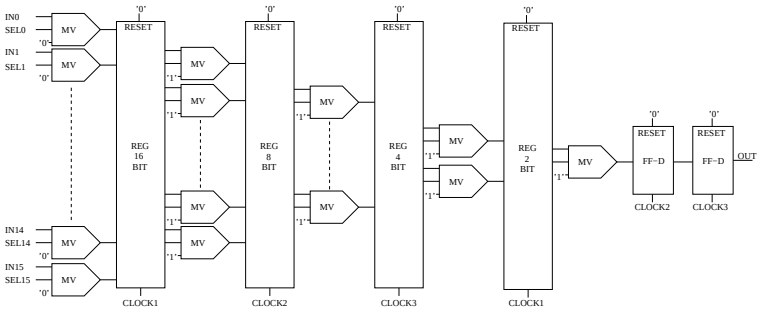



Figure 3.18. A 16 to 1 multiplexer used to select the correct memory output.

AND gates (majority voters with one input connected to '0') are used to select between read and write operations and to set the memory output to '0' during writing operations. Delay blocks are used to synchronize signals to obtain a working circuit.

**Performances**

To evaluate performance the same division algorithm, previously applied to the NCL version of the microprocessor, was applied also to this version of the microprocessor. Waveforms are not reported because are identical to those shown in Figure 3.10, but with a changed time scale.

The performance of the mixed logic processor are shown in Table 3.2. The time required for an instruction execution is slightly smaller, $4.41\mu s$ instead of $5.35\mu s$. The improvement is not so high because in the previous case the NCL memory was a parallel memory so it had not so big an impact in the time balance. The big improvement is in the estimated number of nanomagnets, which is 600K instead of 4M, and the power dissipation which is 6 times smaller.

Table 3.2. Mixed logic microprocessor performances. *M.Vacca et al."Asynchronous Solutions for Nanomagnetic Logic Circuits", ACM Journal on Emerging Technologies in Computing Systems, 2011*

| | NCL | Boolean-NCL |
|---|---|---|
| Istruction execution time [$\mu$s] | 5.35 | 4.41 |
| Area (number of nanomagnets) | $4*10^6$ | $0.6*10^6$ |
| Nanomagnets power dissipation [$\mu$W] | 23.9 | 3.51 |
| Clock power dissipation (Joule Effect) [$\mu$W] | 39.99 | 7.95 |

Results of table 3.2 confirms that the use of a mixed logic solution allows to save a lot of area and therefore to greatly reduce power consumption. The reduction of latency is not so high so circuit speed is still a problem. While these are good results, the overall performance of the circuit are not satisfactory. It is also clear from the analysis in Chapter 1 Section 1.4 that the presence of at least one feedback signal slows down the operations of any QCA circuit implemented in any technology. To maximize the performance the entire microprocessor must be designed with boolean logic but, in this case, two problems arise: Signals synchronization become much more complicated, and an asynchronous like protocol is still needed to handle feedbacks. The innovative solution found is based on the implementation of the circuit using only boolean logic and developing an innovative asynchronous-like protocol to synchronize signals.

### 3.1.3 Full Boolean Logic

The idea behind the full boolean implementation of the microprocessor is based on the discussion on feedback signals presented in in Chapter 1 Section 1.4. The idea is therefore to use a synchronization block placed at the beginning of each loop. This

loop will keep the output constant, and only every N clock cycles a new data will be sampled. The value of N is chosen equal to the delay of the longest loop. The use of this synchronization block, sketched in Figure 3.19, leads to an asynchronous-like protocol. The microprocessor architecture is the same presented before, but now all the blocks are implemented using Boolean logic, and synchronization blocks are used to handle the communication protocol for signals synchronization.

### Asynchronous-like protocol

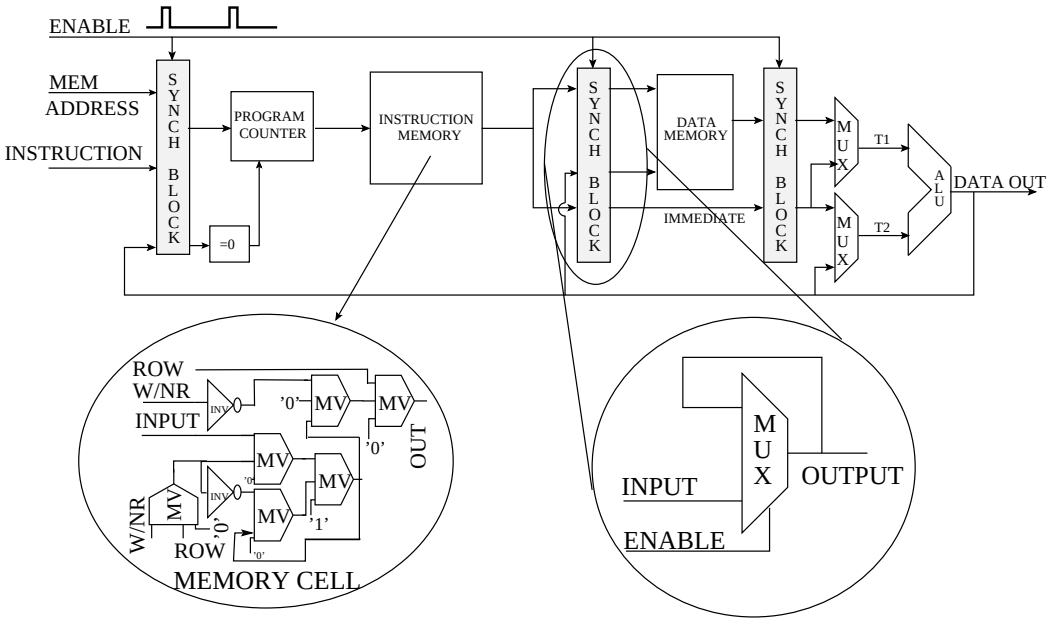



Figure 3.19. Boolean microprocessor architecture. Asynchronous registers are substituted with synchronization blocks (bottom right inset) that realize an asynchronous-like structure. In bottom left detail the boolean memory cell is shown, used in the mixed Boolean-NCL and in the fully Boolean versions of the microprocessor. *M. Vacca et al. "Asynchronous Solutions for Nanomagnetic Logic Circuits", ACM Journal on Emerging Technologies in Computing Systems, 2011*

The synchronization block schematic is shown in Figure 3.19 in the bottom right detail. It is implemented using a multiplexer with the output connected to one of its input. The other input accepts incoming data from outside. The multiplexer normally is in the loop mode. The selected input is its output. In this situation the output will maintain always the same value.

However no new inputs are accepted and the circuit is then frozen in the same state: this is equivalent to a latch in the memory stage. When a time correspondent to the longest loop inside the circuit passed, a new input is sent. But at the same

time a short pulse (ENABLE) is sent to the selection bit of the synchronization multiplexer. This signal travels through the circuit with the input data. When the ENABLE signal reaches the multiplexer, all inputs are already at destination. The pulse allows then the multiplexer to sample the new inputs, that are stored till the next pulse arrives. In this way again an asynchronous communication protocol was implemented, but the whole circuit has a lower complexity as there is not encoding and no handshaking.

As explained before, using NCL logic, a new data can be sent only after a time equal to 4 times the latency of the longest pipe stage, thanks to the NCL communication protocol and the time required for signal propagation. With this solution a new data can be sent after a time equal to the latency of the longest pipe stage. This means that this solution is 4 times faster then the NCL approach. The gain in speed is however higher than 4 times, because the architecture is much more simple than the NCL version. This is an asynchronous system specifically designed for NML (and QCA) technology. It can be used only in this technology because the signals propagation time depends on their layout so it can be known with absolute precision.

### Boolean Program Counter

Figure 3.20 shows the schematics of the boolean program counter. The circuit architecture is similar to its NCL counterpart but simplified, because no loop registers are necessary. Also the "Reset ncl" block can be eliminated. A ripple carry adder is used to generate the next address while a multiplexer allows the selection with an external address. Delay blocks are used to simulate the NML real layout and to synchronize signals.

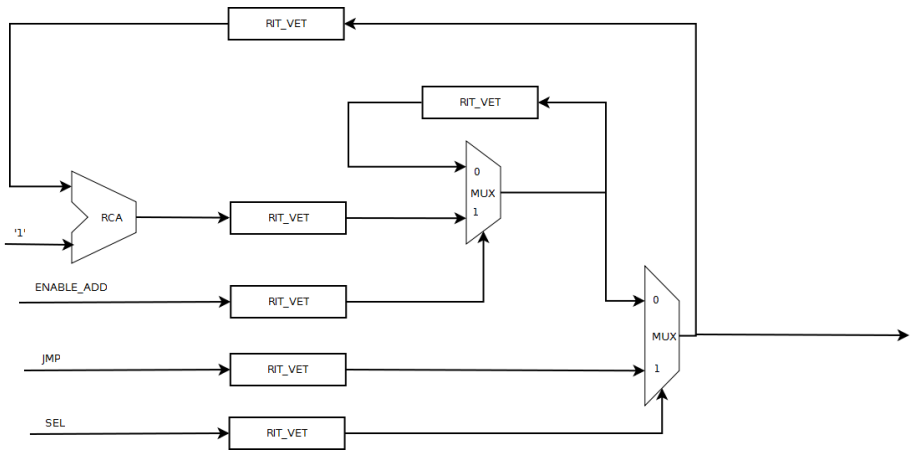



Figure 3.20. Boolean program counter.

### Boolean Alu

Figure 3.21 shows the architecture of the boolean alu. It is identical to its NCL counterpart, where a ripple carry adder is used to perform arithmetic operations and a separated block is used for logic operations. Multiplexers are used to select among the available operations. The memory differs for two details: first the full adder is made using boolean logic, particularly using 3 majority voters (Figure 3.21 on the right), second, delay blocks are used to better emulate NML real layout and to synchronize signals.

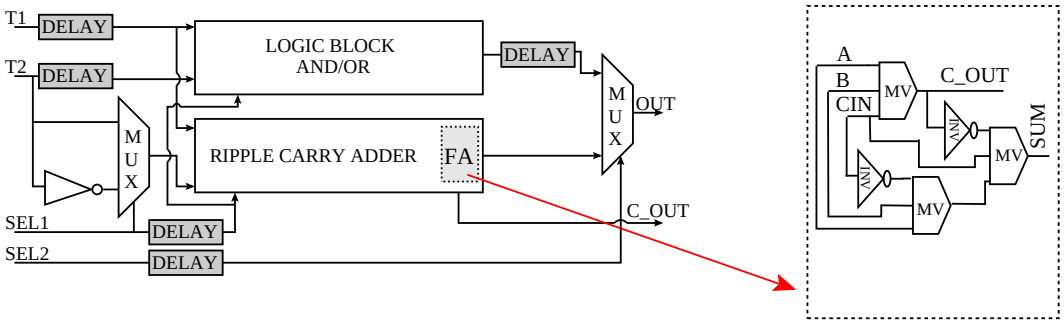



Figure 3.21. Boolean alu. *M.Graziano, M.Vacca et al."Asynchrony in Quantum-Dot Cellular Automata Nanocomputation: Elixir or Poison?", IEEE Design & Test of Computers, 2011*

To better understand what happens with bad signals synchronization, the layout of the alu was intentionally altered, slightly changing the delay of one signals. As it is possible to see from Figure 3.22 there is glitch the generates an error in the computation. This shows how critical is the necessity to achieve a perfect signal synchronization.

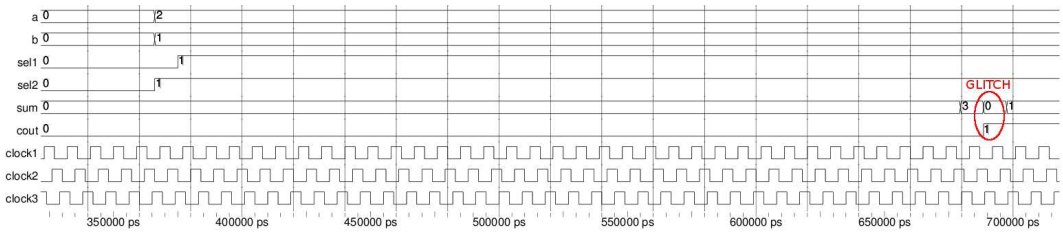



Figure 3.22. Example of glitch generated during alu operations due to bad synchronization. *M.Graziano, M.Vacca et al."Asynchrony in Quantum-Dot Cellular Automata Nanocomputation: Elixir or Poison?", IEEE Design & Test of Computers, 2011*

### Performances

To test the performance of the microprocessor the division algorithm was used again as a benchmark. Figure 3.23 shows the simulation results, which are similar to those shown in Figure 3.10. The difference is that, since with boolean logic no signal encoding is necessary, there are half of the signals with respect to the one shown in Figure 3.10. It is possible to observe a clear boost in the performance, compared to the mixed logic solution and the pure NCL solution. The execution of one instruction requires only $28\mu s$ instead of $194\mu s$ of the mixed logic case, that means about 7 times less. The synchronization pulse is the ENABLE signal shown in Figure 3.23.

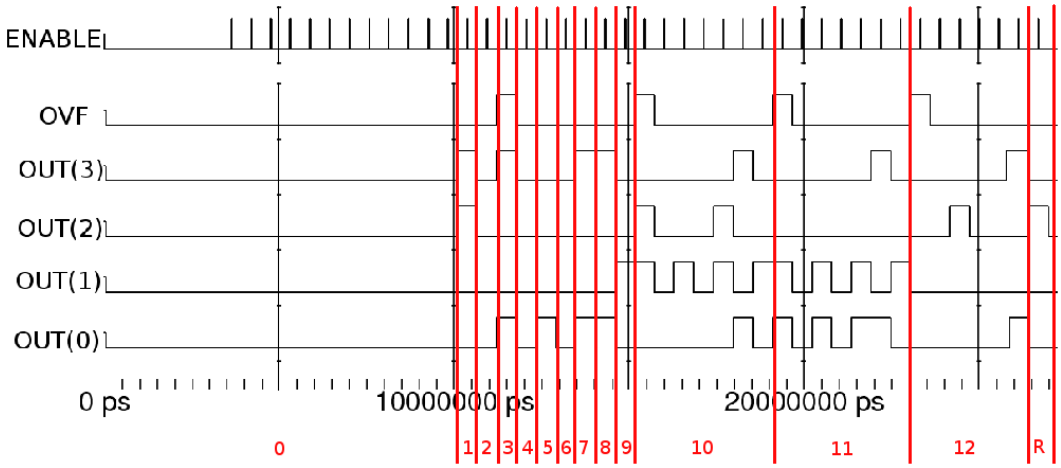



Figure 3.23. Simulation results of the division algorithm executed on the pure Boolean microprocessor. *M. Vacca et al. "Asynchronous Solutions for Nanomagnetic Logic Circuits", ACM Journal on Emerging Technologies in Computing Systems, 2011*

Table 3.3 shows the comparison among the three types of microprocessor. The improvement in terms of speed is about 7 times over the mixed logic solution. This improvement derives from the optimization of the asynchronous protocol and the simplification of the architecture. The number of magnets and therefore their power consumption is 3 times lower, but also the power consumption due to clock losses is reduced by 4 times, mainly thanks to the area reduction.

The conclusions are plain to see: This is the solution that must be followed in the development of NML (and QCA) circuits, because it is the only solution that allows to obtain good enough performance, providing at the same time a simpler way to build circuits. The main problem is that using a full boolean solution the layout of the circuits must be carefully designed to avoid synchronization problems inside

Table 3.3. Microprocessor types comparison. *M. Vacca et al. "Asynchronous Solutions for Nanomagnetic Logic Circuits", ACM Journal on Emerging Technologies in Computing Systems, 2011*

| | NCL | Boolean-NCL | Boolean |
|---|---|---|---|
| Istruction execution time [$\mu$s] | 5.35 | 4.41 | 0.546 |
| Area (number of nanomagnets) | $4*10^6$ | $0.6*10^6$ | $0.2*10^6$ |
| Nanomagnets power dissipation [$\mu$W] | 23.9 | 3.51 | 1.09 |
| Clock power dissipation (Joule Effect) [$\mu$W] | 39.99 | 7.95 | 1.86 |

a pipeline stage. This can be easily obtained using a regular clock zones layout as shown in Chapter 4. However, while this solutions is the best asynchronous solution for NML logic, performance can be still enhanced. A regular clock zones layout (see Chapter 4) allows to obtain an automatic signals synchronization, eliminating therefore the necessity of using asynchronous logic, simplifying the circuit and reducing power consumption and latency. Synchronization problems still remain but in Chapter 5 solutions to synchronize signals in presence of feedback are presented.

# Chapter 4

# Improved Circuits Layout

## 4.1 Enhanced clock zones layout

While the snake-clock solution described in Chapter 1 allows the signals propagation in each direction, which is a mandatory constraint to build any kind of circuit, it presents some major flaws:

- **Wire twisting**. To twist wires they must be placed on different planes. This can be done as confirmed by [2] but the wire section that as a 45 or 90 degrees orientation (Figure 4.1( can be difficult to fabricate.
- **Wasted area**. The area where the wires twist is a forbidden zones where no magnet can be placed, and this means that 22% of the circuit area is wasted.
- **Global inefficiency**. Trying to design complex circuits using this clock layout leads to an inefficient area exploitation. Particularly the width and height of clock zones must be quite big leading to problems in signal propagation due to the very high number of magnets cascaded. Secondly, a vast majority of the area is wasted to route signals to interconnect logic blocks.

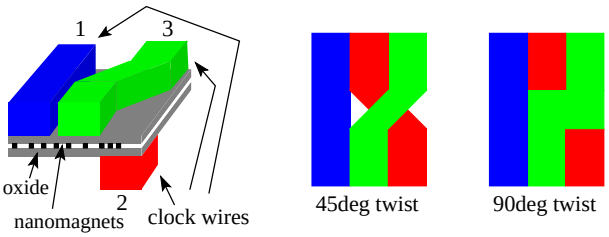



Figure 4.1. Snake clock. Wire twisting can be at 45 degrees or 90 degrees, but this can be difficult to fabricate.

Unfortunately this clock system cannot be completely avoided because it is the only solution that I) can be technologically fabricated and II) allows propagation of feedback signals. A further analysis of QCA technology in general and therefore also NML logic, shows that it is particularly suited for pure combinational circuits. Moreover combinational circuits normally occupy the major part of circuit area. As a consequence the idea is to change the clock zones layout, separating the combinational and sequential parts. The basic idea is shown in Figure 4.2.

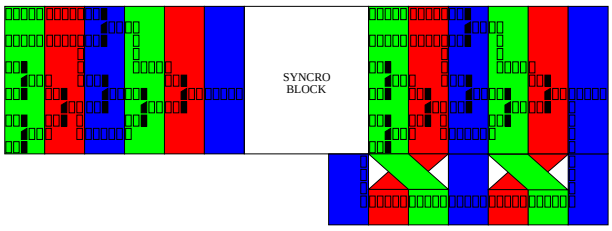



Figure 4.2. Improved circuit layout. Combinational and sequential parts of the circuit are separated. Wire twisting is limited only to feedback signals.

The circuit can be thought as combinational blocks interconnected among them. The important consequence is that the necessity of twist wires is limited only to the areas of the circuit where feedback signals are required, greatly reducing the technological complexity of this solution. Secondly, the wasted area is much smaller and the circuits compactness is enhanced. As shown in Figure 4.2 a synchronization block (described in Chapter 3) can be used at the end of the feedback signals, to make sure that signals have time to propagate back. The synchronization block can be avoided (further enhancing the circuit compactness) using the solutions described in Chapter 5, which describe how to synchronize signals in case of feedbacks.

## 4.2 Combinational logic circuit structure

An example of layout of a combinational block is shown in Figure 4.3. It is based on clock zones made by parallel strips. This layout is chosen according to the experimental results shown in [2], where parallel wires placed on different planes are used to generate the clocking magnetic field. This layout is particularly suited for circuits with a dataflow structure, with inputs coming from one side and output generated at the opposite side. Circuits have therefore a “tree” shape as can be seen from Figure 4.3.

The width of each clock zones must be chosen according to the maximum number of magnets that can be cascaded without having errors in the signal propagation process. According to [39] this value is 5, which means that a maximum of 5 elements

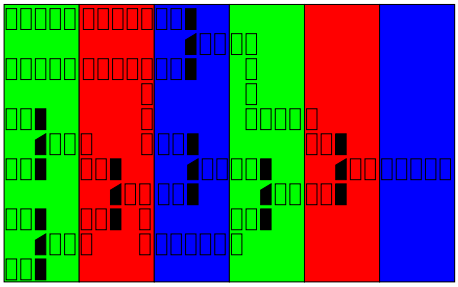


Figure 4.3. NML Combinational circuits layout. This layout is technologically feasible and particularly adapted to dataflow logic.

can be cascaded to avoid errors in the signal propagation. This is a very limiting constraint that leads to a width for the clock zone of 4 magnets. The maximum length for horizontal wires (horizontally aligned magnets) inside a clock zone is therefore 4 magnets, while the maximum length for vertical wires (vertically aligned magnets) inside a clock zone is 2 magnet. With this choice the critical path (i.e. the maximum number of magnets cascaded) is 5 (Figure 4.4.A). As demonstrated in [3] to help signal propagation when magnets are vertically aligned, helper blocks (Figure 4.4.A) are necessary. This blocks are made by magnets with an aspect ratio lower than 1, that are always in the RESET state. They are used to keep the magnetization of vertically aligned magnets when the magnetic field is applied stable and to lower the value of the magnetic field required to force these magnets in the RESET state. Chapter 7 describes what happens if helper blocks are not used.

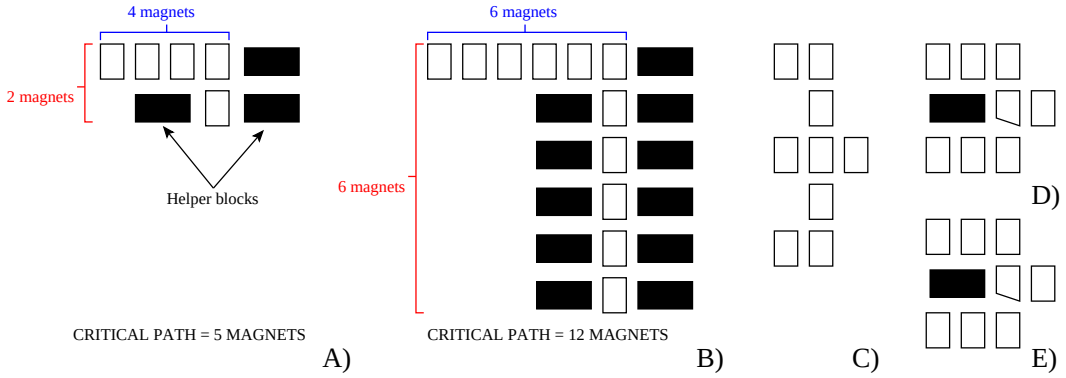



Figure 4.4. Constraints related to the clock zones layout. Helper blocks are used to help signal propagation in vertical direction. A) More constraining case: Critical path of 5 magnets. B) Relaxing of some constraints: Critical path of 12 magnets. C) Modified majority voter. D) AND gate. E) OR gate.

A critical path of 5 magnets is a strong constraint that has a serious impact on

circuit architectures and also on the technological fabrication of clock wires, that must be quite small. However the theoretical results of [39] seems to be negated by many experiments, for example like in [4], where long chains of magnets propagate signals without errors. This is probably due to a limitation in the LLG equation, which describes the magnetodynamics of micron-size magnetic system, and therefore is unable to perfectly model a nanometric-size single domain magnetic system. As a consequence more relaxed constraints can be used, particularly 6 magnets both for horizontal and vertical wires (Figure 4.4.B), that leads to a critical path of 12 magnets. This is a choice that reasonably allows to avoid errors, removing at the same time some of the circuital limitations. The critical path has also an influence on the clock frequency. Lesser magnets in the critical path means higher clock frequency, so choosing a critical path of 5 magnets is better from the performance point of view. Also with this clock zones layout the majority voter shape must be changed, as shown in Figure 4.4.C. As described in Chapter 7 simulations show that this structure do not work, unfortunately experimental results show that this type of majority voter works correctly [4], so this gate can be used to build circuits. Instead AND/OR gates (Figure 4.4.D.E) works perfectly with this clock zones layout, both in simulation than in experimental evidences, so it is better to use this gates as main logic gates.

Figure 4.5.A shows an example of circuit layout based on this clock zones layout. The circuit is a LDPC decoder, a particular type of decoder used in high speed WI-FI communication systems. In Figure 4.5.B it is instead indicated the detailed layout of the CMP block, which is the basic logic block of this architecture. Magnets have 60 nm width and 90 nm height. The schematic shown in Figure 4.5.A gives a general idea of the circuit layout of the whole decoder. The circuit has a very elongated shape. This is due to the chosen clock system, which favors the signal propagation in the horizontal direction and penalizes the vertical signal propagation. If a long vertical interconnection is required a “stair-like” signal propagation must be used (Fig. 4.5.C), increasing the width of the circuit. The “stair-like” vertical signal propagation is an important consequence of this clock zones layout. The elongated layout of the circuit, with a balanced placement of the blocks, is therefore chosen to minimize the wasted area due to vertical signal propagation.

## 4.3 Application of CMOS architectures to NML logic

After fixing the clock zones layout, and therefore the global circuits structure, it is possible to investigate which kind of circuit architectures that are best suited for NML technology and the chosen clock zones layout. To perform this analysis a 32

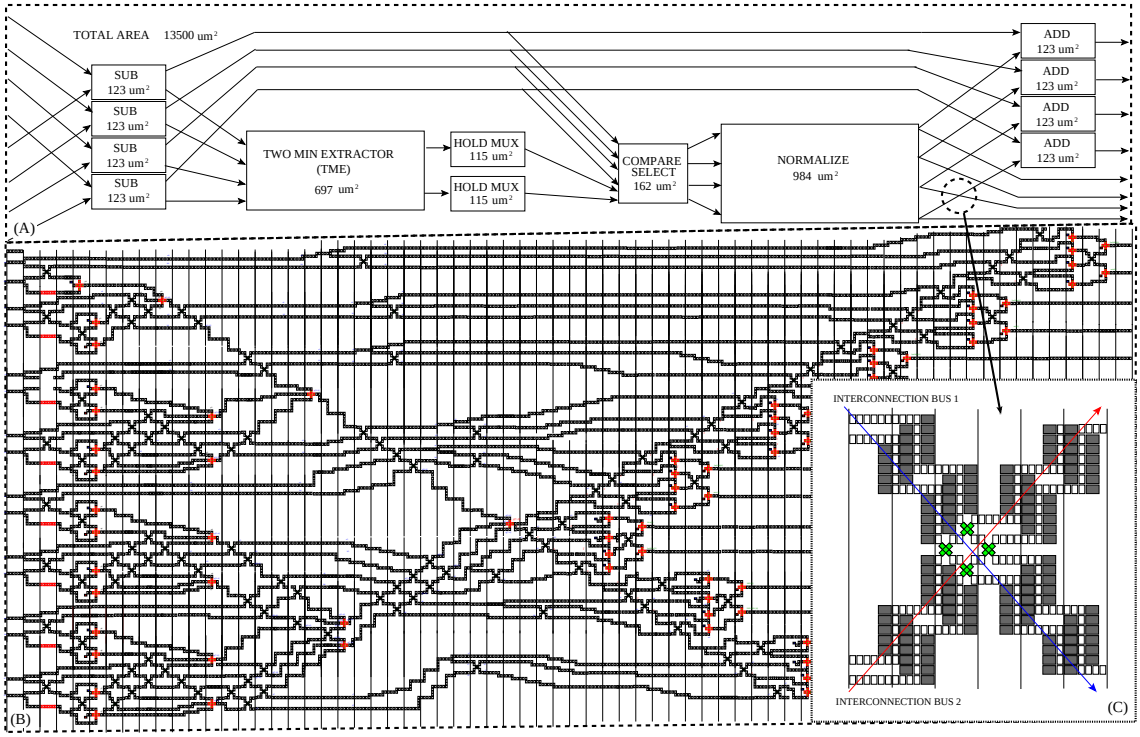



Figure 4.5. A LDPC decoder for wireless applications. The layout is based on straight wires for the generation of the clock field. This layout was theoretically and experimentally demonstrated for Magnetic QCA [2]. B) CMP block layout. C) A detail on vertical interconnection wires. Due to the layout limitations vertical signals follow a "stairs-like" propagation. Stabilizer blocks are used to improve the reliability in vertical signal propagation [3]. *M.Awais et al."Quantum dot Cellular Automata Check Node Implementation for LDPC Decoders",IEEE Transaction on Nanotechnology, 2013*

bit adder was chosen. This choice was done because the adder is the basic logic block of any logic circuits. The parallelism of 32 bits is chosen because most results can be highlighted only if a complex enough circuit is used. The adder was implemented first with a structure similar to the Pentium 4 adder and then using th most simple architecture, the Ripple Carry adder.

### 4.3.1 Pentium 4 adder

The Pentium 4 adder uses a sparse tree to implement the carry generation network, while eight 4 bits Carry Select adders are used to calculate the sum. Its correspondent NML implementation is shown in Figure 4.6. The detail shows the layout of one of th full adders used in the calculation of the sum. The circuit layout is based on

the clock structure presented above. The sparse tree used for the carry generation is the same of the CMOS Pentium 4 adder, while for the sum calculation Ripple Carry Adders are used instead of the original Carry Select adders. As can be seen the circuit area is quite big, and it is mainly due to the clock zones layout constraints. Particularly in this case clock zones are 4 nanomagnets width and the critical path is made by only 5 magnets. Moreover only AND/OR gates are used. This choice assures that, if this circuit will be fabricated, it will work in every condition. The area occupied by the nanomagnets is 30-40% of the total area.

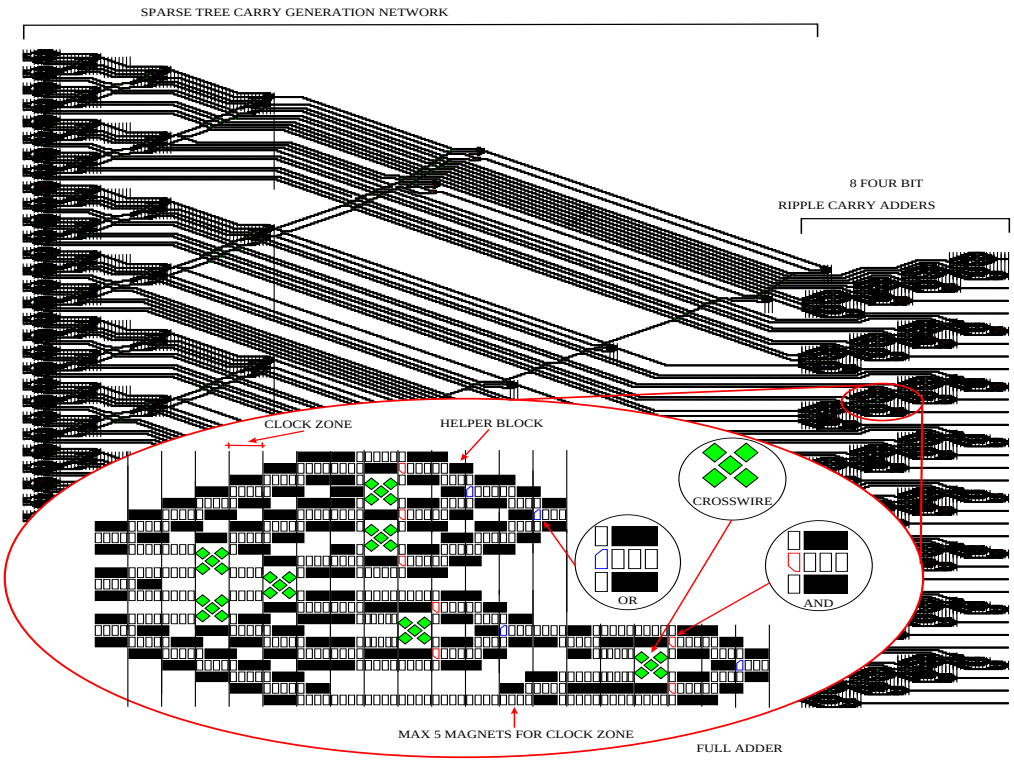



Figure 4.6. NML 32 bits pentium 4 adder. A sparse tree carry generation network is coupled with eight 4 bit ripple carry adder. *M. Vacca et al. "ToPoliNano: A synthesis and simulation tool for NML circuits", International Conference on Nanotechnology, 2012*

### 4.3.2 32 bit Ripple Carry Adder

The Ripple Carry Adder is the most simple of the adders. In CMOS it has the smallest area but the highest delay. In NML logic things are different. In this technology circuits are intrinsically pipelined and the clock frequency is a technology

constraint independent from the circuit architecture. Changing the architecture only changes the circuit latency. The layout of a 32 bit Ripple Carry adder is shown in Figure 4.7. Full adders are aligned along a diagonal, in order to connect the carry out of each full adder to the carry in of the next full adder. Inputs and outputs are carried by long wires, in order to have all of them at the same x-coordinate. This circuit can be further optimized if it is part of a more complex circuits. Others components can be placed in the area that is now occupied by input/output wires. The full adder used is the same used in the case of Pentium 4 adder and it is shown in the left detail of Figure 4.7. A more compact circuit can be used to implement the full adder, using majority voter instead of AND/OR gates. This full adder is shown in the right detail of Figure 4.7. It is based on [4] and it uses a 12 magnets critical path solution. Although this solution is partially experimental demonstrated, the first solution is more reliable from the errors point of view.

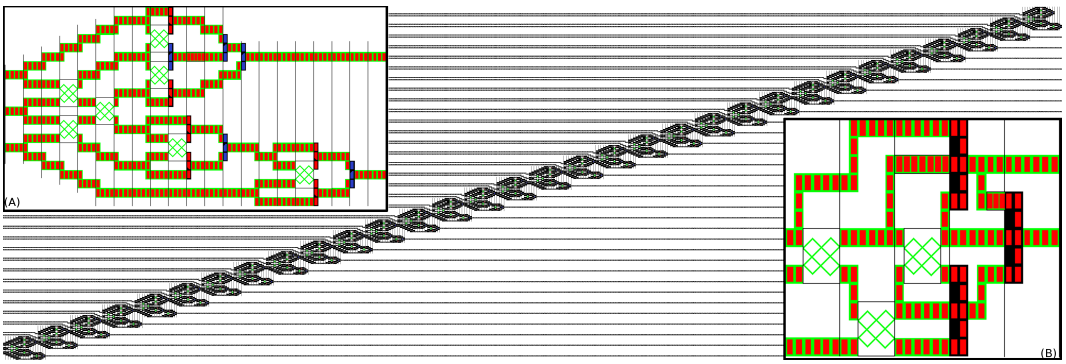



Figure 4.7. 32 bits ripple carry adder. Two different types of fulladders are shown, one which uses AND/OR gates and a clock zone with a width of 4 nanomagnets, a second one which uses Majority Voters [4] and a clock zone with a width of 6 magnets. *M. Vacca et al. "ToPoliNano: A synthesis and simulation tool for NML circuits", International Conference on Nanotechnology, 2012*

### 4.3.3 Comparison

Figure 4.8 shows the comparison between the two adders. The Pentium 4 adder is shorter but its height is bigger than the Ripple Carry Adder. Apparently the Pentium 4 adder is the best solution, the most area efficient, because the total area is smaller than the total area of the Ripple Carry adder. However this is not true: If these adders are used as part of a more complex circuit, the Ripple Carry adder will be smaller than the Pentium 4 adder. Two important points must be considered. First signals propagate vertically using a "stair-like" path. This means that the longer the vertical interconnection is, the larger will be the circuit. Since the height

of the Pentium 4 adder is bigger, the total circuit length will be increased considering interconnection wires. Secondly, inserting the adder in a complex circuit, the Ripple Carry adder area which is now occupied by input/output wires, can be used to place other components. This means that the global circuit area can be optimized to keep into account clock layout characteristics, therefore the final Ripple Carry adder area will be much smaller than the Pentium 4 adder area.

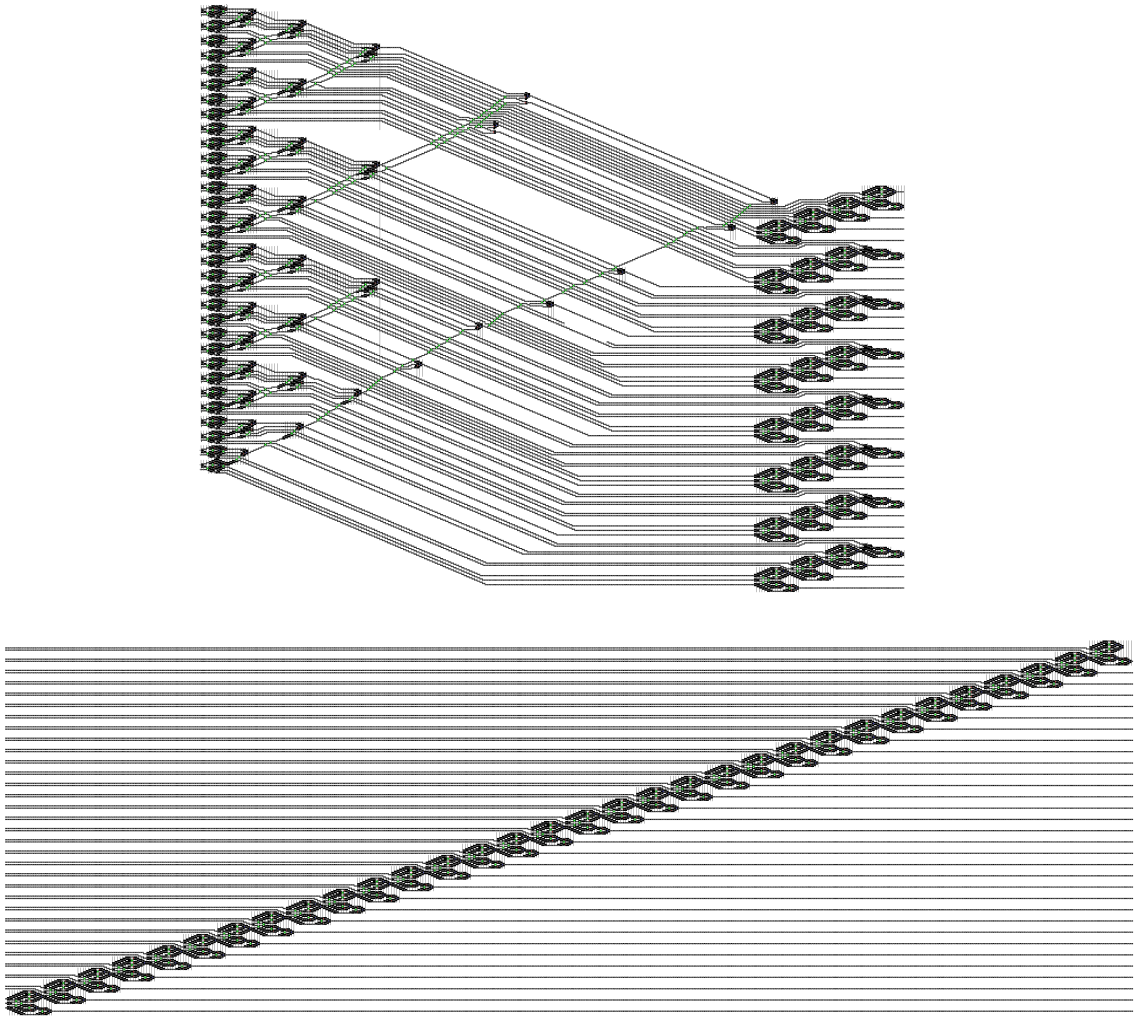


Figure 4.8. Comparison between the P4 adder and the ripple carry adder. The ripple carry adder area is only slightly higher than the P4 adder. With this clock zones layout the simplest architectures are favored.

From this comparison some important consequences can be extrapolated.

- **Circuit architectures**. Since the clock frequency is not related to the circuit

architecture but only to technological constraints, the only things that changes with the architecture of the circuit is the area, and therefore the circuit latency (which is important in case of sequential circuits). In case of NML logic, differently from CMOS, the simplest of the architecture is the optimum solution because it allows to reach the best overall performance.

- **Parallelization**. Increasing the level of parallelization of the circuit means to increase its height, therefore the width of the circuit increases and so does its latency.
- **Nanomagnets area**. Only the 30-40% of the total area is occupied by nanomagnets, that means that a 60-70% is wasted.
- **Interconnections area**. Of the area occupied by the nanomagnets 99% is used for interconnection wires, and only 1% for logic gates. This is due to the fact that with this technology up-to-now circuit are confined on only one plane. It is important to underline that area, in NML, means higher latency but also higher power consumption. So the wasted area due to interconnections contributes to reduce the advantages of this technology.

The important result is that to build complex circuits is necessary to find solutions to improve global interconnections. While local interconnections can be implemented with nanomagnets, longer wires requires necessary other means to propagate the information. The use of systolic arrays (see Chapter 5) with they regular layout can help, but this does not remove the need of more efficient ways to propagate signals at longer distances. Another important consequence is that, to build complex circuits, ways to extend this technology on multiple layers (like CMOS) must be investigated.

## 4.4 NML interconnections improvement

To improve interconnections the most obvious solution is to translate the magnetization of a magnet into an electric signal. This signal can travel along copper wires and then can be converted again in the magnetic domain. Other solutions can be based on other emerging magnetic technologies, like Domain walls, Spin-Torque coupling and Spinwaves.

### 4.4.1 Input and output interfaces

To use electric interconnections two types of interfaces are required: An input interface that can influence the first magnet and an output interface that can convert the magnetization of the last magnet into an electric signal. This two interfaces can also be used as general input and output of the NML chip.

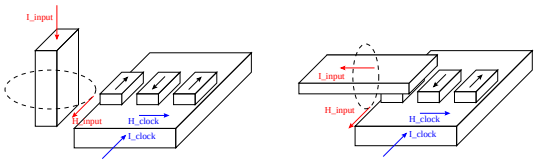



Figure 4.9. Current flowing through wires can be used to generate a magnetic field used to influence an input magnet.

A first idea of possible input interface can be to use the magnetic field generated by a current flowing through a wire placed near the input magnet. Figure 4.9 shows two possible examples of this approach. In the first case a wire perpendicular to the plane is used, while in the second case a wire parallel to the plane and to the short side of the magnets is used. Both approach works, but the fabrication of this structure can be tricky.

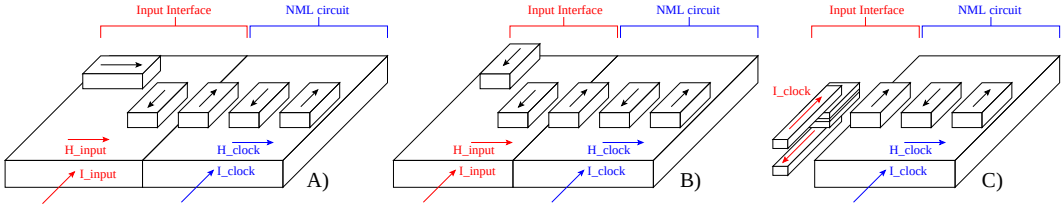



Figure 4.10. A) Classic input interface. B) Improved input interface. C) MTJ input interface.

In literature, for example like in [4], a different solution is used. This solution was the first technique ever used and is shown in Figure 4.10.A. A magnet rotated of 90 degrees is placed near the upper or lower border of the first magnet. When the magnetic field is applied to the circuit, all the magnets are forced in the RESET state. When the magnetic field is removed the magnetization of the magnet rotated of 90 degrees will still be parallel to its long side, while the other magnets will start to rotate according to this magnet. This solution can works both in case of externally applied magnetic field than in case of magnetic field generated by a clock wire. The advantage of this solution is that the geometry of the wire used as input is identical to the geometry of the wires used for clocking, simplifying the fabrication of the circuit. The direction of the magnetic field generated depends on the direction of the current (see Figure 4.10.A), therefore controlling the direction of the current allows to write a logic '0' or a logic '1'. An improved solution here developed is shown in Figure 4.10.B. The magnet used to force the input is not rotated of 90 degrees but it is diagonally aligned. With this configuration if the magnetic field is applied in the right direction, when it will be removed magnetization will rotate

down, in the '0' logic state. If the magnetic field is applied in the left direction, when it will be removed magnetization will rotate in the up direction, in the '1' logic state. The reason of this behavior is related to how the magnetic flux lines interact among them and it is explained in chapter 7. Also in this solution controlling the direction of the current allows to control which value will be written in the input magnet. Similarly to the classic solution also in this case the geometry of the input wire is similar to the geometry of the clock wires. The advantage of this solution is that the required value of magnetic field is lower therefore the power dissipated is reduced. Figure 4.11 shows the low level simulation of this structure, when a magnetic field is applied in the right direction (write '0', Figure 4.11.A) and when a magnetic field is applied in the left direction write '1', Figure 4.11.B). As it is possible to see the solution works correctly.

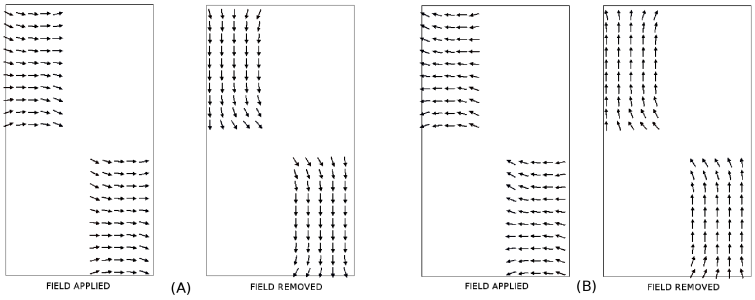



Figure 4.11. A) Low level simulation of input '0'. B) Low level simulation of input '1'.

However both these solutions can be used only at research level. For the commercial development of this technology more reliable solutions must be used. The perfect candidates for this purpose are the MagnetoTunnel Junctions (MTJ). MTJ are multilayered structures made by one insulating layer placed between two layers of magnetic materials. The thickness of this layers is quite low, in the range of 1-2 nm. The lower of the magnetic layers is made by an hard magnetic material, which magnetization cannot be changed, while the upper layer is made by a soft magnetic material. The upper layer changes its magnetization according to neighbor elements as happens normally in NML logic. There are two advantages in this structure. First the value of the magnetization of the soft magnetic material can be changed using a current which flows through the MTJ. Second the resistance of an MTJ depends on the relative orientation of the magnetization of the two layer. The resistance (considering magnets of 50x100 $nm^2$) switches from 2000 ohm to 2600 ohm, depending on the relative orientation of the magnetization vector of the two magnetic layers. This structure can therefore be successfully used both as input structure, using a current flowing thought the MTJ and controlling the direction of the current to obtain the desired logic value (see Figure 4.10.C), but also as an

output structure, because to measure the value of resistance means to measure the logic states of an MTJ. This structure is the most suited solution for input/output interfaces for NML logic because it is very reliable and it is already experimentally demonstrated [23][40][41]. This structure is commonly used in Magnetic RAMs, which are already available at commercial level.

### 4.4.2 Electric interconnections

Since both input/output interfaces are available, it is therefore possible to use electric interconnections in NML logic. The most important point is that the circuit used to read/write magnets must use a limited number of transistors. This is required for two important reasons: If there are too many transistor the entire purpose of NML logic will be wiped out, secondly the power dissipation of the electric interconnection must be kept as small as possible. The main reason because electric interconnections should be used is to reduce the circuit latency but, most importantly, the power dissipation due to the wasted area necessary for long NML interconnections. The trade off between the power consumption of a long NML wire and the whole electric interconnection (input interface + electric interconnection + output interface) determines how many of this interconnections can be integrated inside a NML circuit. In [40] a reading circuit for MTJ is shown. While this system works quite well is too complex to be used for the purpose of building electric interconnections, since 10 transistors and an amplifier are required. Simpler solutions are necessary.

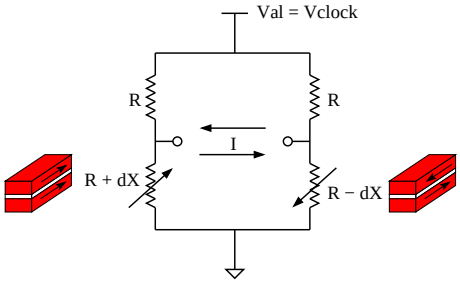



Figure 4.12. Simplest example of electric interconnection, a resistive H Bridge is used to read the value of MTJ and to drive another magnet.

Using MTJ junctions the value of magnetization is represented by a variation of resistance and it must be translated in a variation of current direction, since to write a '0' or a '1' in an MTJ stack means to flow a current in opposite directions. This can be a little tricky if the aim is to keep the number of transistor as small as possible. To change the direction of the current a bridge circuit is necessary, a H Bridge or an Half Bridge. The simplest solution can be the use of a resistive H Bridge (Figure 4.12). In this schematics two neighbor MTJ are read at the same

time. Two neighbor MTJ (i.e. magnets) are always in an opposite state, that means that one has the maximum value of resistance while the other one has the minimum value. As a consequence if the value of the constant resistors in the bridge is chosen exactly equal to the MTJ resistance mean value, the current flowing in the middle of the bridge will change its direction according to the state of the two neighbor MTJ used as sensors. If these two magnets are in the "01" configuration current will flow in a direction, if they are in the "10" state current will flow in the opposite direction. If another MTJ is connected in the middle of the bridge, it can be written to '0' or to '1' according to the value of the output MTJ, implementing therefore an electric interconnection.

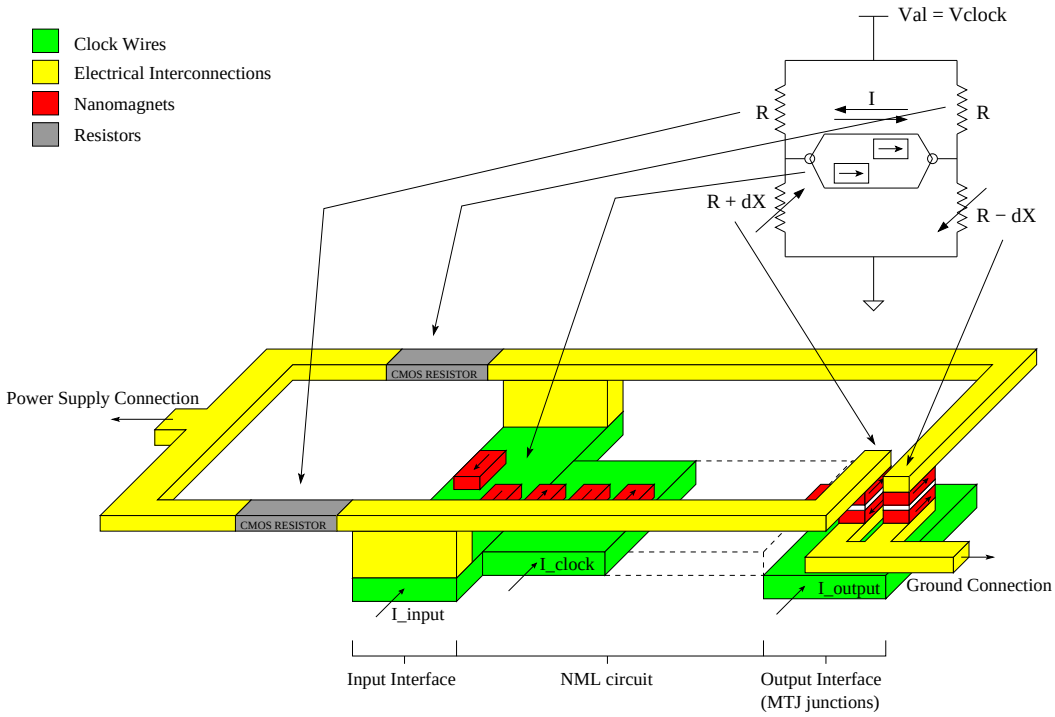



Figure 4.13. Example of complete electric interconnection system used for a feedback signal.

Figure 4.13 shows an example of 3D layout of an electric interconnection used for feedback signals. In this case the input structure is not based on an MTJ but on the structure shown in Figure 4.10.B, however also in case of an MTJ input the structure will be similar. As it is possible to see the structure is quite simple, and simple is always a good thing because it simplifies the fabrication of the circuit. Unfortunately this structure does not work. The main problem is that the current that must flows through the contacts in the middle of the bridge, which is used to write the logic '0' or '1' in the input magnet, this current flows also through the sense

MTJ that are used to read the magnetization value. This current is strong enough to change the magnetization of the sense MTJ altering therefore the measurement.

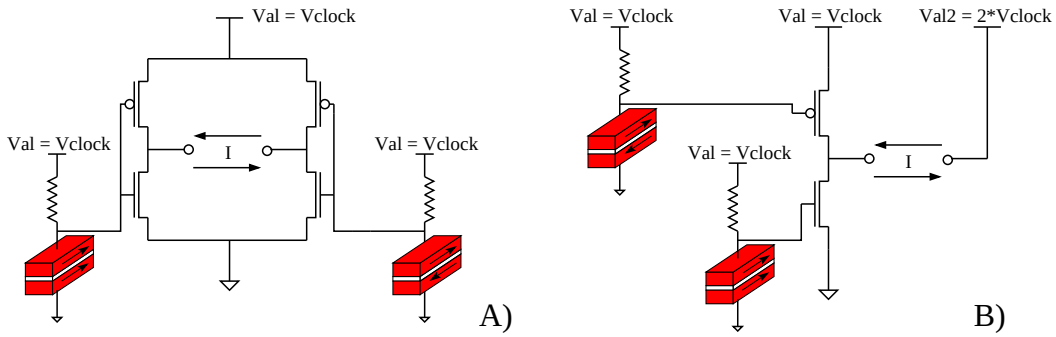



Figure 4.14. Alternative electric interconnection circuits. A) Full bridge. B) Half bridge.

Improved (and more complex) circuits must be used, particularly it is necessary to use a certain amount of transistors to isolate the reading part from the writing part of the circuit. Two possible schematics are shown in Figure 4.14. In Figure 4.14.A a classic H Bridge made by 4 transistors is used to change the direction of the current. The transistors of the bridge are driven by two simple resistances serially connected (one is the MTJ). Depending on the value of the MTJ the voltage on the transistor gate can or cannot be enough to activate the transistor. One side of the bridge is driven by one MTJ while the other side is driven by its neighbor MTJ which is always in the opposite state. This solutions requires at least 4 transistors for each electric interconnections. It is possible to use only 2 transistors with an Half Bridge circuit, which one example is shown in Figure 4.14.B. Two voltage levels are used $Val$ and $Val2$, with $Val2 = 2 \cdot Val$. One transistor is connected between the first voltage $Val$ and the contact that is used to drive the current on the input magnet, while the other transistor is connected to the ground and the same contact. The other contact is connected to the second voltage $Val2$. This two transistors are again driven by two neighbor MTJ, so they are always in opposite states, one open and one close. The current flows therefore in two opposite directions depending on the value of the sense MTJ. In all this circuit the voltage is always taken from the voltage used to generate the clock, so that the interconnection is active only during the relative clock phase to minimize power consumption. These are only two possible circuits that can be used for electric interconnections, their main disadvantage is that they are based on resistances, which value must be carefully tuned, but it mean also that there will be a static power consumption.

### 4.4.3 Applications

Electric interconnections can have many important application inside NML circuits. They can be used for feedback signals, reducing the length in terms of clock cycle of the loop improving the circuit throughput (see Chapter 5). They can be used to replace long interconnection wires, leading to a Standard Cells approach for NML logic, where compact clusters of combinational logic are connected through electric interconnections. It is possible to imagine further applications, because this approach allows to control the clock signals of part of the circuit depending on the magnetization of a magnet, therefore it is possible, for example, to shut down an entire part of the circuit if it is not used. It is also possible to develop a logic-in-memory approach. If those magnets are clocked they are used as normal cells. If the clock signal is removed they act like a memory. Another possibility is to integrate Magnetic RAM inside the circuit, and this can be a great advantage of this technology.

However the analysis of this kind of structures can be quite tricky, because there are no physical simulators that can simulate all the involved effects. Since an MTJ is a variable resistance a SPICE model can be built. At the physical level there are simulators that can handle separately different parts of this structure, but not together. The idea is therefore to develop a framework that allows to use different simulators together. Let's suppose that the effect of the magnetic field generated by a current flowing through a wire on nanomagnets must be accurately verified. It is possible to simulate the magnetic field generated by a current using COMSOL Multiphysics, but COMSOL cannot simulate the magnetodynamics. From the other side OOMMF (the most used of the low level magnetic simulators) can simulate the magnetodynamics but it cannot simulate a magnetic field current-generated. It is possible to think at a framework which runs a simulation of COMSOL Multiphysics, calculating the spatial distribution of the magnetic field generated by a current flowing through a wire. This magnetic field can be exported by the framework on a file that can be used as input by OOMMF, to verify the dynamics of the circuit, without loosing information on the spatial distribution of the magnetic field. Such a framework can be extended easily to other simulators, and it is mandatory if accurate simulations of systems, like electric interconnections in NML circuits, that are based on the interaction of so many different physical phenomenons.

### 4.4.4 Full magnetic interconnections

The interconnection problem is related to the intrinsic pipelining of NML circuits. The level of pipelining is very high due to the limited amount of magnets that can be placed inside a clock zone. As a consequence, if interconnections are too long (Figure 4.15.A), the propagation delay can be very high. However there are other ways to

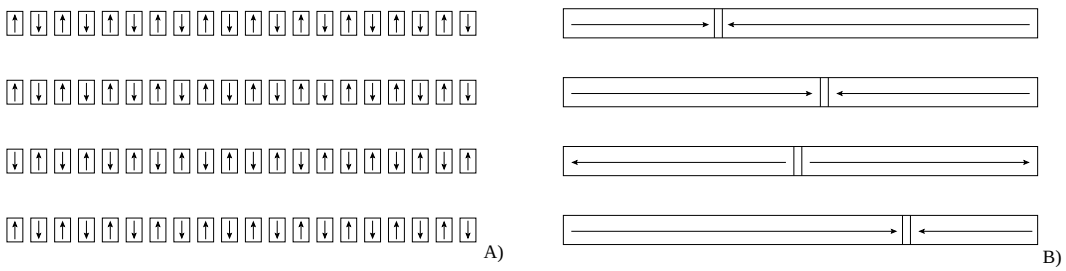



Figure 4.15. Magnetic Wires. A) Nanomagnet wires. B) Domain wall wires

propagate magnetic signals, that can be exploited to build high speed magnetic interconnections. Probably the best solution for magnetic interconnections is the use of domain walls. Domain wall interconnections are essentially (Figure 4.15.B) long strips made of ferromagnetic materials. In the relaxed state, this long strips are uniformly magnetized along the length of the strip. If the magnetization is locally forced in the opposite state at the beginning of the strip through an external mean, two regions will be created inside the strip. This two regions (Figure 4.15.B) will have opposite magnetization. As a consequence a new region will born among them, where magnetization normally lies out-of-plane. This region is called Domain Wall and it moves along the magnetic strip propagating therefore the information through the circuit. Domain walls can move at very high speed (more than 1500 m/s) and are therefore a good candidate to implement interconnections in a full magnetic circuit, greatly reducing the delay due to long interconnections.

# Chapter 5

# Architecture improvements

## 5.1 Feedback signals

The use of a multiphase clock system generates an intrinsic pipelined architecture, where every group of 3 consecutive clock zones has a delay of 1 clock cycle and is therefore equivalent to a CMOS register. This is true not only for NanoMagnet Logic, but also for any other implementation of Quantum dot Cellular Automata technology (molecular, metallic or semiconductor). What changes is the placement and the number of clock zones, but each one of this technologies will have an intrinsic pipelined behavior.

### 5.1.1 Throughput reduction

The intrinsic pipelined behavior has important consequences on the circuit, particularly it causes a huge drop of throughput in case of sequential circuits [42] that include loops in their combinational flow. This problem is a severe limitation for a so promising technology. To understand this problem a simple circuit, shown in Figure 5.1, is used as an example. It is a simple serial adder where the output is connected to one of the inputs. The delay of the loop is in this case 4 clock cycles. Since NML is intrinsically pipelined it is possible to send a new data every clock cycle, to obtain maximum throughput. Unfortunately this is true only in case of a combinational circuits but not in case of sequential ones. In this case for example, if a data ($A$) is sent at the first clock cycle (Figure 5.1.A) and immediately after one clock cycle another data ($B$) is sent (Figure 5.1.B), the results is not ($A$+$B$) as expected. The reasons of this lie behind the propagation time required for the first output to reach the adder input.

In this case the output needs 4 clock cycles to reach the adder input. To synchronize signals and obtain the correct result, a new data must be sent not after 1 clock cycle but after 4 clock cycles (in this example). This can be seen from Figure

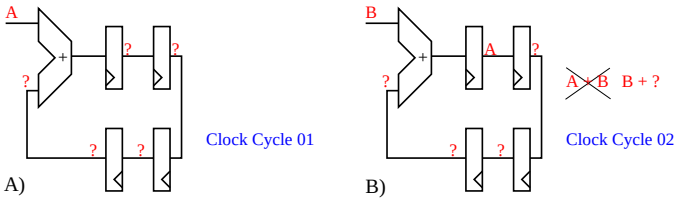



Figure 5.1. Effect of loops in intrinsic pipelined technologies. A) Sending a data and B) immediately after a clock cycle sending a new data, lead to the wrong result, because the previous result had not time to propagate back.

5.2. In Figure 5.2.A a data ($A$) is sent to the input of the adder and it is kept frozen for other 3 clock cycles, as shown in Figure 5.2.B, Figure 5.2.C and Figure 5.2.D. Only at the 4th clock cycle a new data ($B$) is sent to the input of the adder. As it is possible to see from Figure 5.2.E signals are perfectly synchronized, because ($B$) is sent to the adder input exactly when ($A$) reach the other adder input. The results is therefore ($A+B$) as expected.

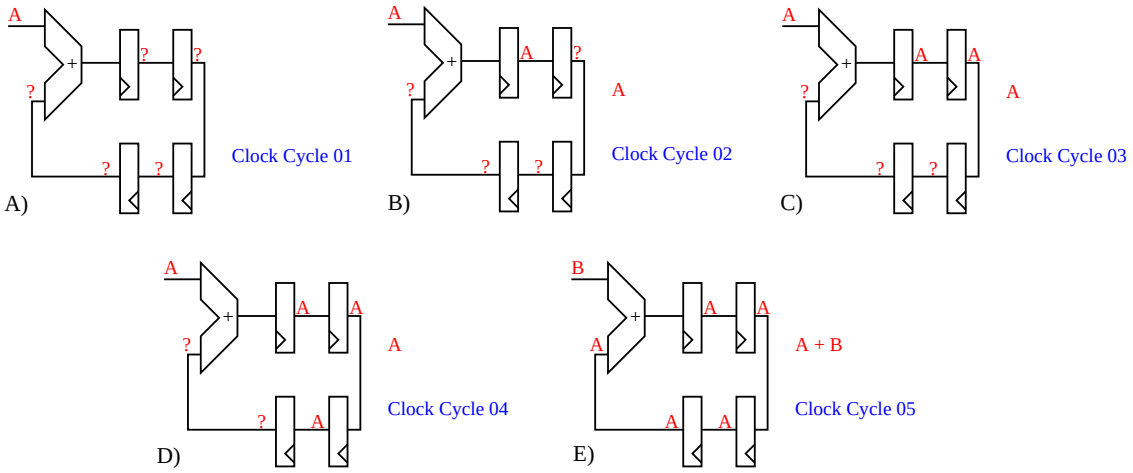



Figure 5.2. Effect of loops in intrinsic pipelined technologies. A) Sending a data and B) C) D) keeping the input constant for N clock cycles, E) allows to obtain the correct result, because the data had time to propagate back.

The fundamental point is that, when there is a feedback in a NML (or QCA) circuit, to synchronize signals and to obtain correct results input data can be sent to the circuit not every clock cycle but every N clock cycles. N is the delay of the longest loop inside the circuit, and in case of NML (or QCA) technology can be very high, hundreds of clock cycles. As a consequence the circuit throughput is reduced by N times. In other words the circuit is N times slower. This is a very

limiting factor for this technology. For example, let's suppose to build a hypothetical molecular QCA microprocessor. Molecular QCA are interesting because they can potentially reach extremely high clock frequencies (1 THz). If compared to a 1GHz CMOS microprocessor the molecular processor is expected to be 1000 times faster. But, if a loop of 1000 clock cycles is present inside the circuit the throughput of the molecular microprocessor will be the same of the CMOS processor, wiping out all the advantages of this technology.

### 5.1.2 Throughput maximization: Data Interleaving

The impact of pipelining in sequential circuits is a well known problem in CMOS, and it is one of the reasons why pipelining is not so heavily exploited in CMOS technology. For example, superscalar microprocessors suffer of performance drop in case of JUMP instructions during the execution of an assembler program. JUMP instructions are particular instructions that makes the program jump to a particular memory location. The address of this jump location must be calculated by the circuit itself, which has a pipelined structure. As a consequence the calculation of the jump address requires many clock cycles, where no other instructions can be executed. To solve this problem jump prediction techniques coupled with instructions reordering techniques are used. This techniques only partially solves this problem but it is not possible to always reach the maximum throughput. This fact helps to understand how much important the problem of pipelining in sequential circuits is, also in CMOS technology.

Unfortunately this problem is much worse in case of NML (and QCA) technology. In CMOS the level of pipelining is something that can be controlled by the designer changing the circuit architectures, changing the number of registers. In NML (and QCA) pipelining is intrinsic to the technology itself, it can be reduced (see Section 5.1.3) changing the circuit architecture, but it cannot be completely eliminated. Moreover the level of pipelining in NML (and QCA) technology can be extremely high (hundreds or thousands of clock cycles). It is possible to think to apply techniques like jump prediction and instruction reordering also to these technologies, but they can only help to reduce the problem.

To solve this problem in NML (and QCA) technology more radical solutions must be used. The solution here developed is called *Data Interleaving*. It is a rather simple technique that allows to reach the maximum throughput exploiting parallelism. The idea is shown in Figure 5.3 where the same adder with the output connected to one of its inputs is used as an example. 4 operations are executed in parallel on the same circuit, $(A+B+C)$, $(D+E+F)$, $(G+H+I)$ and $(L+M+N)$. At the first clock cycle (figure 5.3.A) the first data of the first operation $(A)$ is sent. At the second clock cycle (Figure 5.3.B) a new data is sent but it is not the second data of the first operation $(B)$, but the first data of the second operation $(D)$. In this

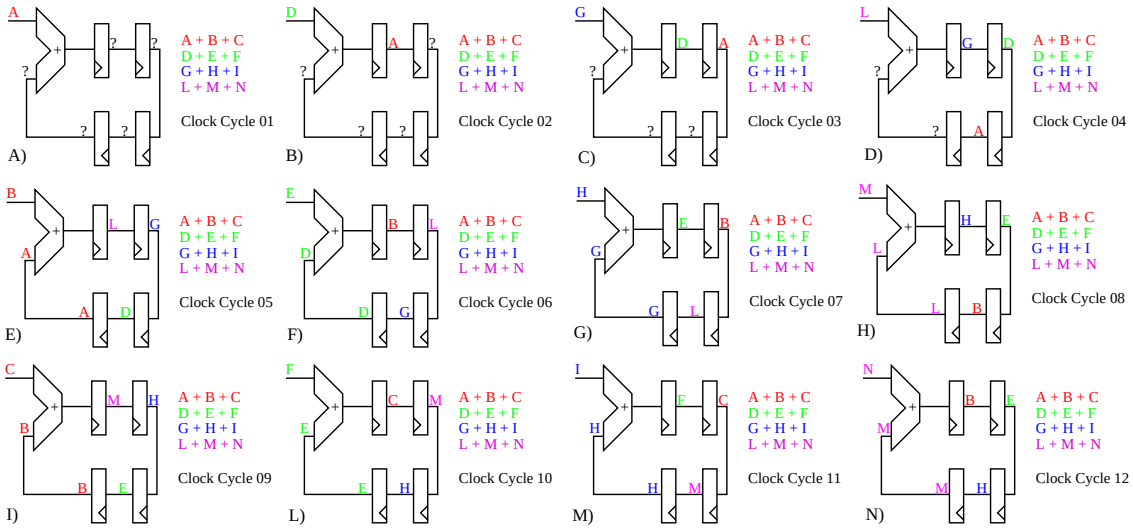



Figure 5.3. Data interleaving. N operations are executed in parallel. Every clock cycle a data of a different operation is sent, achieving perfect synchronization and maximum throughput.

case, differently from the case shown in Figure 5.1.B, the results is correct because $(D)$ is not dependent on $(A)$. At the third (Figure 5.3.C) and forth (Figure 5.3.D) clock cycles the inputs sent are respectively the first data of the third operation $(G)$ and the first data of the forth operation $(L)$. At the fifth clock cycle a new data of the first operation $(B)$ is sent to the adder input. As it is possible to see from Figure 5.3.E signals are perfectly synchronized, because the previous data of the first operation $(A)$ has reached the other adder input. In the next clock cycles (Figures from 5.3.F to 5.3.N) at every clock cycle the approach is repeated, sending every time a different data of a different operation. As it is possible to see this technique achieve perfect signal synchronization but at the same a new data is sent every clock cycle reaching maximum throughput. *Data Interleaving* is a very simple technique which solves the problem of intrinsic pipelining in NML (and QCA) technology if loops are present inside the circuit. The drawback is that N operations must be executed in parallel, where N is delay in clock cycles of the longest loop inside the circuit. Since N can be extremely high in these technologies this technique cannot always be applied.

### 5.1.3 Loops length reduction

Since the interleaving technique cannot be always applied it is important to rearrange the circuit layout reducing the length of the loops inside the circuit. A 2 bits Multiply and Accumulate (MAC) unit (Figure 5.4 in the upper-right detail) is used

as an example. This circuit is based on a multiplier with the output connected to an adder. The output of the adder is connected to its second input. This unit is a very important circuit that constitutes the core circuit of Digital Signal Processors (DSP), circuits specialized in the signal analysis. This circuits assume a particular meaning because, as it will be seen in the discussion, signal analysis is one of the most suitable applications for QCA technology.

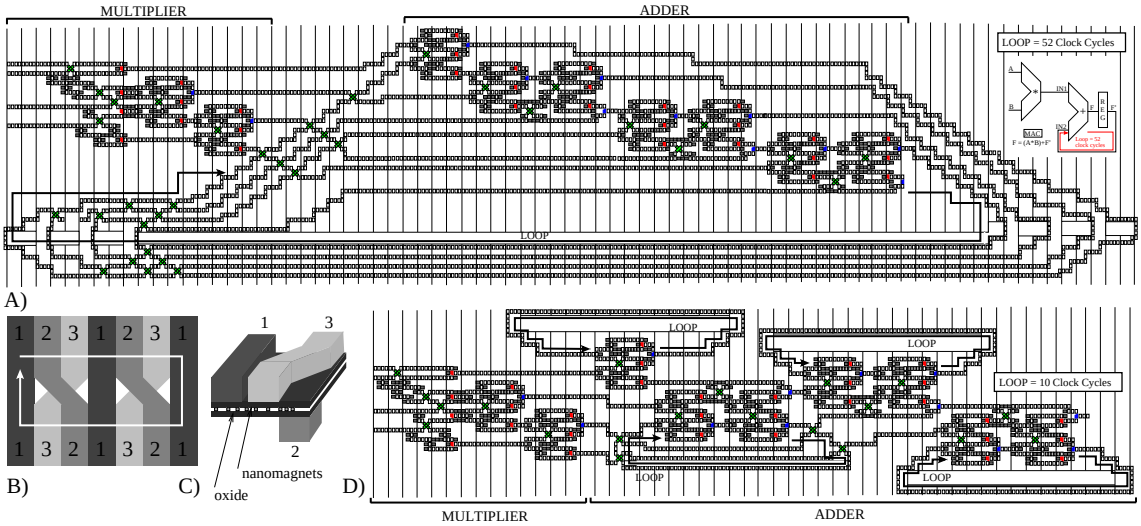



Figure 5.4. MAC detailed layout. The layout uses clock zones made by parallel wires [2], while for feedback it is adopted the solution proposed in [5]. Circuits are made using AND/OR gates [6] that best suit this kind of clock zones layout. A) Direct mapping of the circuit schematics. The longest loop has a delay of 52 clock cycles. B) Top view of the clock zones layout to allow feedback signals propagation. C) 3D view of the clock wires where the current must flow to generate the magnetic field [5] [2]. D) Circuit layout with loops optimization. The delay of the loop is reduced to 10 clock cycles.

Figure 5.4.A shows a possible NML layout of the 2 bits MAC unit. The clock zones are organized in parallel strips, which is a layout based on the clock solution proposed in [2] while AND/OR gates [6] are used as basic logic blocks. Feedbacks are possible locally changing the sequence of the clock zones, as shown in Figure 5.4.B. To obtain this results, wires, where the current that generates the magnetic field flows, are alternatively placed above and under the magnets plane, as first proposed in [5] and then in [2] and then twisted (Figure 5.4.C). Magnets cannot be placed in the region where wires are twisted [5]. This layout leads to a loop delay of 52 clock cycles. Circuits can be arranged for example as shown in Figure 5.4.D. The loop delay is considerably smaller, only 10 clock cycles, it is independent from the bit number of the MAC unit and also more compact.

### 5.1.4 Signals synchronization with feedbacks

The intrinsic pipelining coupled with loops originates more problems of signal synchronization inside complex circuit. For example what happens when there are multiple loops inside the same circuit? If those loops are not nested there are no problems, the longest loop inside the circuit will determine the level of interleaving necessary to obtain maximum performance, but loops can have different lengths. Instead if loops are nested their length must be the same, otherwise signals will not be synchronous. Figure 5.5 shows two different types of nested loops, in both cases the delay of both loops in terms of clock cycles must be identical.

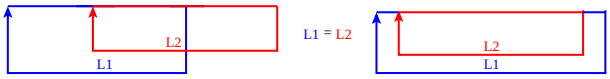



Figure 5.5. Nested Loops. To synchronize signals their length must be exactly the same.

There are situations, in CMOS, where pipelining is exploited for doing operations where inputs are sampled at different time steps. An example is shown in Figure 5.6. In this example an adder calculate the sum of 4 inputs. One input is the output of the previous sum calculated one clock cycle before, while the other 3 inputs are the same input $A$ sampled in different time steps. Particularly the first input is the signal $A$ delayed of 2 additional clock cycles, while the second input is always the signal $A$ delayed of 1 additional clock cycle.

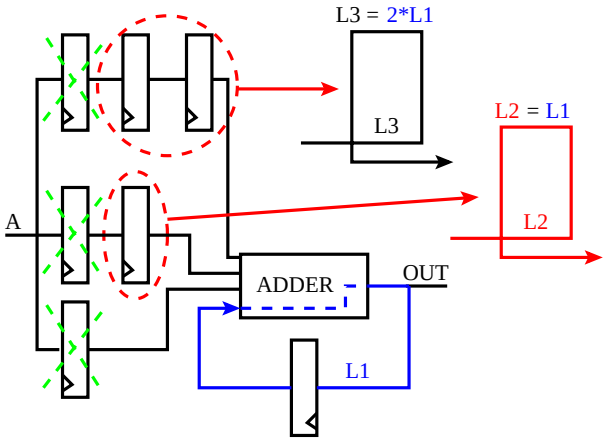



Figure 5.6. Complex signals synchronization. If a loop is present inside the circuit, every additional register, which is not present in all the input paths, must have an equivalent delay equal to the delay of the loop.

An additional delay of 1 clock cycle means that the signal sent one clock cycle before is considered. When the correspondent NML circuit is designed this additional delay must be equal to the delay of the longest loop. In NML (or in general QCA) loops are intrinsically pipelined, therefore a new input must be sent every N clock cycles. Also using interleaving this situation does not change. It is true that with interleaving a new data is sent every clock cycle, but they are part of different operations. A new data of the same operation is sent only every N clock cycles (this can be better understood looking at Figure 5.3), also in case of interleaving. So if a signal in the CMOS circuit has an additional delay of 1 clock cycle, it represents the previous data that was sent to the circuit. But in this case the previous data was sent N clock cycles before. As a consequence additional CMOS registers must be mapped in NML as additional delays equal to the delay of the longest loop. The situation can be understood looking at Figure 5.6. The three registers that are common to the three input signals can be not considered when the NML circuit is designed. The additional register on the second input must be translated in a delay equal to the delay of the loop. Considering the circuit layout described in Chapter 4, the only way to model such delay is to use a NML wire that makes a loop which as the same length of the longest loop of the circuit. The second input has two additional registers, therefore they must be mapped into a loop with a delay equal to 2 times the delay of the longest loop. Summarizing, it is important to understand that in NML (or QCA), also using interleaving, a new data is sent every N clock cycles, so, if the algorithm requires the use of the signal sampled one clock cycle before, a delay of N clock cycles must be inserted.

### 5.1.5 Loop Unrolling

Since loops are so troublesome in NML (and QCA) technology, another possibility is to eliminate them using the technique called *Loop Unrolling*. The basic idea behind this technique is to duplicate hardware and exploit pipelining to do operations on the same data in different time steps.

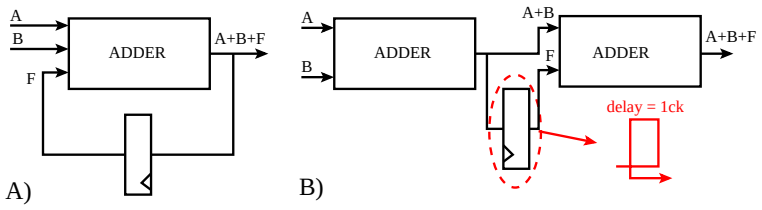



Figure 5.7. Loop unrolling to completely remove loop inside the circuit.

The basic principle is explained in Figure 5.7. The example circuit (Figure 5.7.A)

is an adder which calculates the sum of two external inputs, $A$ and $B$, and the output of the adder itself obtained one clock cycle before. This loop can be unrolled like shown in Figure 5.7.B. The adder is duplicated, the first adder calculates the sum of the two external inputs, $A$ and $B$. The output of the first adder is then duplicated in two branches: The first branch is connected directly to the input of the second adder, while the second branch is delayed of 1 clock cycle and then sent to the second adder. As it is possible to see from Figure 5.7.B the output of the second adder is exactly the same of the adder of Figure 5.7.A. The delay is again obtained using a wire with a loop, as previously done. This technique cannot be always applied but it allows to physically eliminate loops from the circuit, generating huge benefits in NML (and QCA) technology. The disadvantage is that hardware duplication leads to an increase in the circuit area. In case of Figure 5.7 one 3 inputs adder is replaced by two 2 inputs adders. The total area is increased of 50% in the best case, however considering interconnections the area overhead will be higher.

## 5.2 Systolic arrays

As described in Chapter 4 in any NML circuit most of the area is filled by interconnection wires. Medium and long interconnections occupy a lot of area because this technology is confined to only one plane. As discussed in Chapter 4 electric interconnections are a possible way to improve signals propagation in long wires and also to reduce the length of loops (see Section 5.1.3).

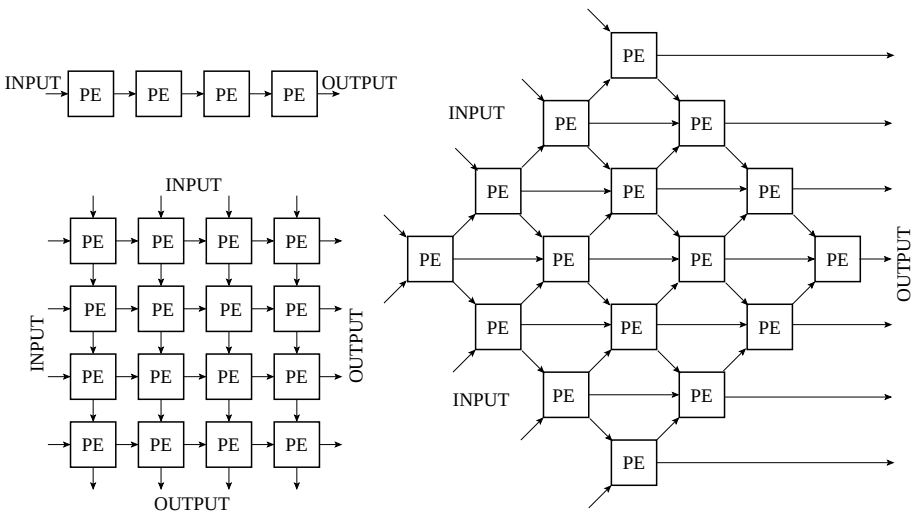



Figure 5.8. Different possibilities for systolic arrays.

However it is also important to design circuits avoiding long interconnections, and

this can be done exploiting systolic arrays [43]. Systolic arrays are circuits made by a network of processors, called processing elements (PE), which rhythmically compute and pass data through the system. In this circuits only local interconnections are present and are used to connect logic elements inside every processing element, and to connect nearby processing elements. Systolic arrays have an extremely regular layout and are also parallel architectures, so they are perfectly suited for NML (and QCA) technology. Also interleaving can be applied quite successfully to these architectures [44]. However systolic arrays represents only a partial solution to the problem of interconnections in NML (and QCA) technology.

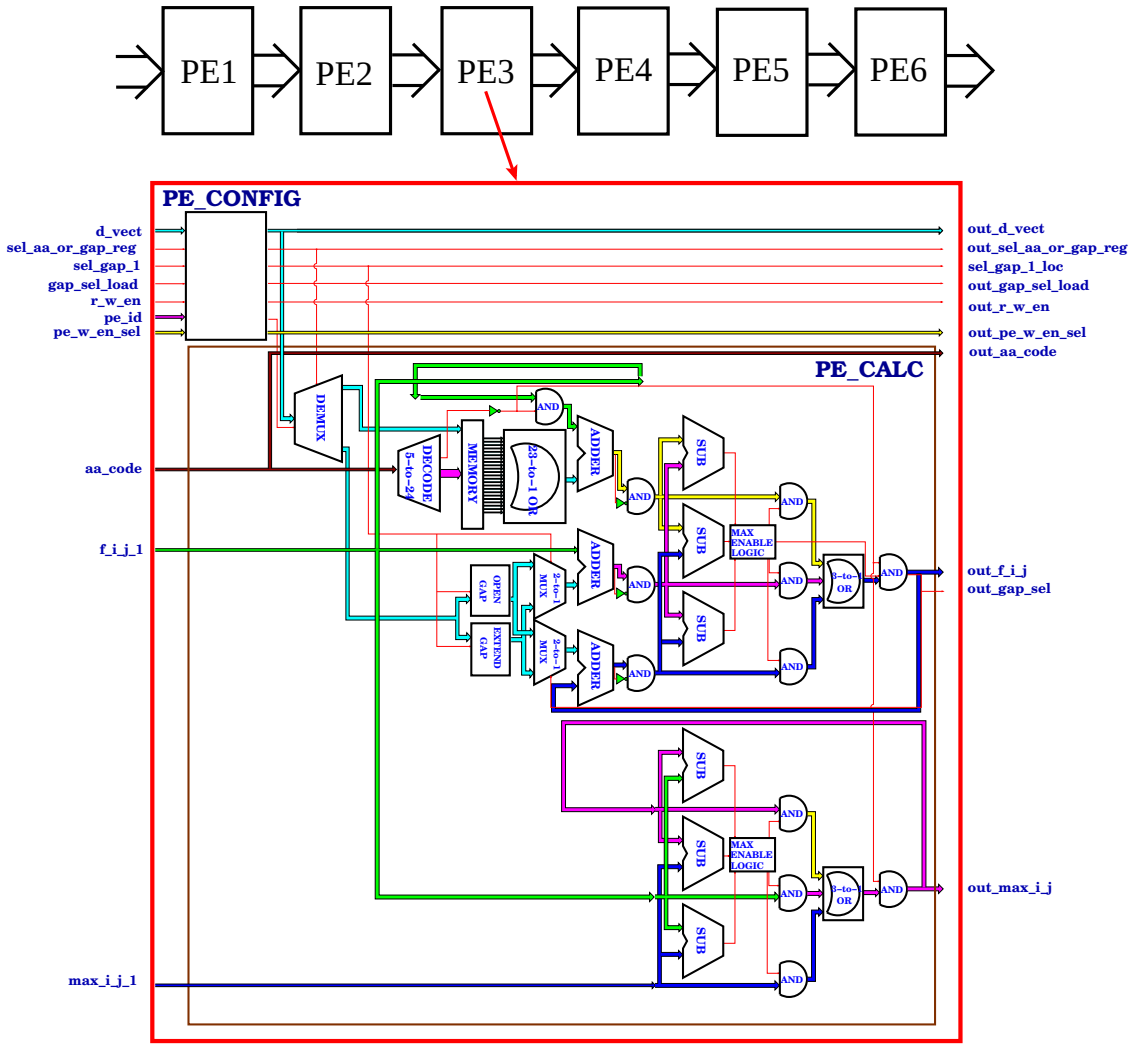



Figure 5.9. Processing Element of the Smith-Waterman implemented in NML with a systolic array architecture [7].

As described in [7], where a complex systolic array architecture implemented in NML logic was studied, when the processing elements has a complex structure all the architectural problems of interconnections and signals synchronization are still present. Figure 5.9 shows the processing elements of the architecture described in [7]. The architecture is the systolic array implementation of an algorithm called "Smith-Waterman", which is used to find local alignment sequences, in long chains of amino acids that compose proteins. The systolic array is made by a linear series of processing elements. While this is a systolic array, the processing element contains many complex components, like adders. There is still a problem of interconnections inside the processing element, that is testified by the fact that the longest loop inside it has a delay of 200 clock cycles. This means that to reach the maximum throughput 200 operations must run in parallel. This is not a problem in this case because there are normally a lot of sequences that must be verified and the length of the database is also extremely high.

This characteristics put in evidence which kind of applications is best suited for NML (and QCA) technology, which applications can exploit the true potential of the technology: *Massively Parallel Data Analysis applications*, like the one shown in [7], are the ideal candidate application for this technology. However it must be pointed out that systolic arrays alone are not enough to limit the interconnection problem. The architecture of the processing elements must use components with a simple and regular architecture, like shown in Section 4.3.

### 5.2.1 Programmable Systolic Array

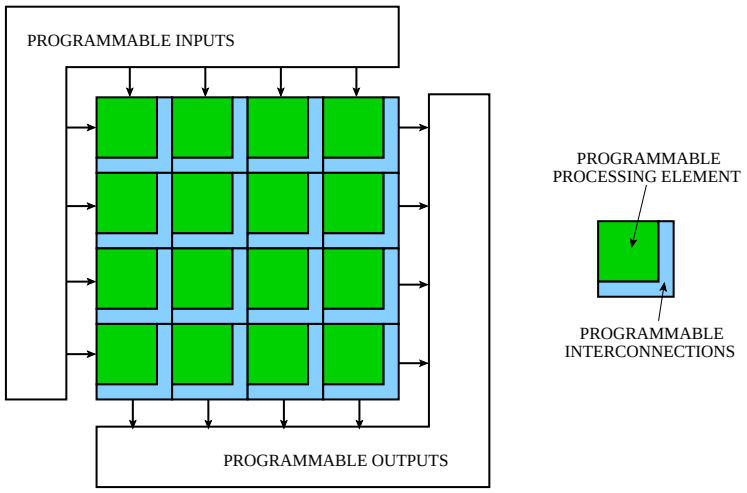



Figure 5.10. Programmable systolic array structure.

As previously said, systolic arrays are the best kind of architectures for NML (and QCA) technology. However systolic arrays are a class of architectures that are best suited for specific tasks. They are also specialized architectures, not made for general purpose computation. This was the limiting factor of the technology, that never had a good commercial success.

To solve this problem the solution here proposed is to develop a programmable systolic array, similar in the structure to FPGA. The circuit is shown in Figure 5.10. The idea is to use a network of processing elements that can be programmed to obtain any desired logic function, like happens in FPGA. Processing elements are regularly placed on the chip surface and are interconnected by programmable interconnection blocks. Depending on how interconnection blocks are programmed any kind of systolic array structure (see Figure 5.8) can be obtained. The structure is completed by a set of programmable input/output pins. The whole circuit as a structure similar to FPGA, therefore it merges the application specificity of systolic arrays with the flexibility of FPGA greatly enhancing the commercial possibilities of NML (and QCA) technology.

# Chapter 6

# ToPoliNano: a synthesis and simulation tool for NML circuits

## 6.1 Motivation

To analyze NML circuit normally low level finite element simulator like OOMMF [45] and NMAG [46], are used. They are based on the Landau-Lifshitz-Gilbert (LLG) (equation 6.1) equation which describes the dynamic behavior of micromagnetic systems. The results obtained are very accurate but the computational power required is very high so only small circuits can be simulated and the simulation is also very slow.

$$\frac{\partial M(r,t)}{\partial t} = -\gamma M(r,t) x H_{eff}(r,t) - \frac{\alpha\gamma}{M_s}[M(r,t) x (M(r,t) x H_{eff}(r,t))] \tag{6.1}$$

For complex circuits VHDL models can be successfully used [42]. The keypoint (as described in Chapter 2) is to describe a CMOS circuit that behaves exactly like its NML counterpart. The core of the model is the behavior of one clock zone, at which a clock signal is applied: At every clock cycle a new data is accepted. This is the same behavior of a register so, registers can be used to simulate the clock zones behavior while ideal logic gate with no delay are used to model the logic behavior of the circuit (Figure 6.1). In this way it is possible to fast describe and simulate very complex circuits, but the results are inaccurate because too many information related to the layout are lost.

For generic QCA circuits a small tool for automatic layout generation of combinational circuits (QCALG [47] is available. This tool is based on QCAdesigner [48], however nothing is available in the case of NML, for simulations and for physical design. So the situation is the one presented in Figure 6.2. From one side there is a

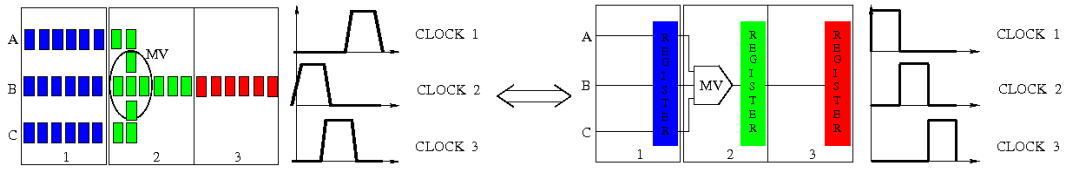



Figure 6.1. Vhdl modeling.

low level simulation, accurate but slow, and from the other side there is the VHDL high level simulation, inaccurate but fast.

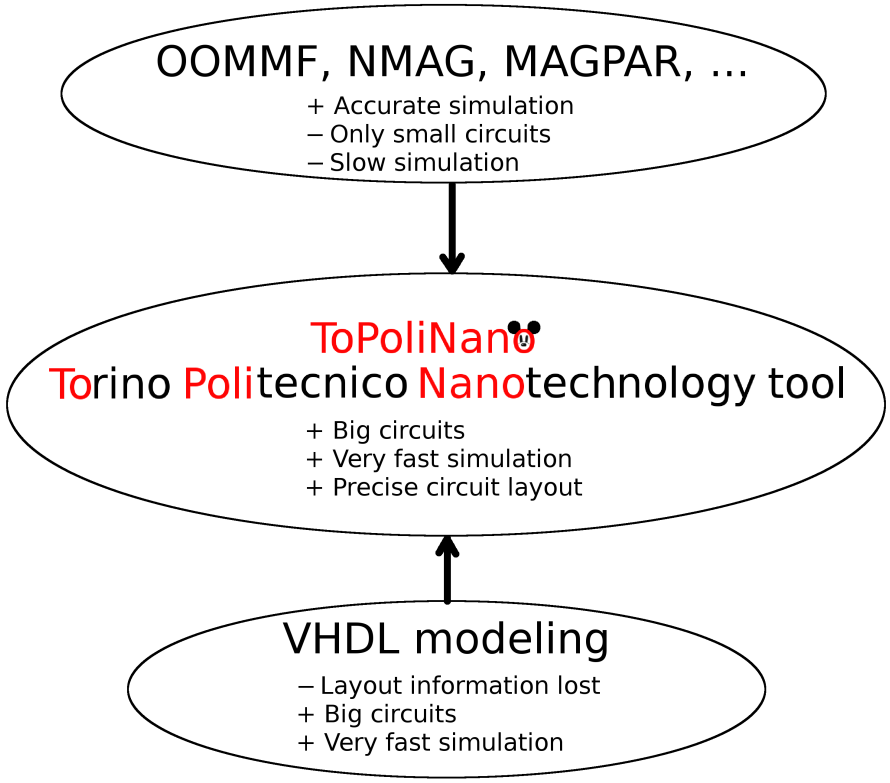



Figure 6.2. NML simulation.

ToPoliNano (Torino Politecnico Nanotechnology tool) is the missing link between these two levels. It is a tool designed to emulate the Top-Down design flow of CMOS circuits. This means to describe circuits using a generic VHDL description, to automatically generate the circuit layout and to fast simulate it using a simplified switch model like the one used in CMOS digital simulator. As a consequence ToPoliNano allows the fast description and simulation of very complex circuits, without losing

layout information, obtaining therefore accurate results.

The philosophy behind ToPoliNano is quite simple: **Easy design and fast simulation of complex circuits made with different nanotechnologies**. That means:

- **Easy design**: Circuit are described using VHDL language and the layout will be automatically generated.
- **Fast simulation**: To use a simplified switching model based on low level simulations and experimental results.
- **Different nanotechnologies**: It has a modular structure to allow the development and integration of simulation engines and layout generators for different emerging technologies.

Figure 6.3 shows a screenshot of the program which shows the Graphical User Interface (GUI). ToPoliNano is written in C++ and it will use full parallel computation, both on CPU and GPU. The choice of C++ as a language, was done due to its nature of Object Oriented Programming Language. The use of Object and Child Classes allows for the higher possible flexibility and modularity. ToPoliNano works on every operating system.

It is an ongoing work, so some parts are still to be completed. It is based on the work presented in [49].

## 6.2 General structure

Figure 6.4 shows the modules and the general design flow of ToPoliNano.

- **Logic Synthesizer.** The logic synthesizer analyze the VHDL code, which can contain a generic circuit description, and maps this circuit on the technology library available, that means, in case of NML logic, inverters, majority voters, AND/OR gates.
- **Parser.** The parser takes as input the output file generated by the logic synthesizer and creates a graph. This graph represents the in-memory description of the circuit.
- **Place & route.** The Place&Route automatically generate the circuit layout starting from the in-memory description generated by the parser. It is divided in two parts, the Placer which physically place logic gates and the Router which connect logic gates with wires.

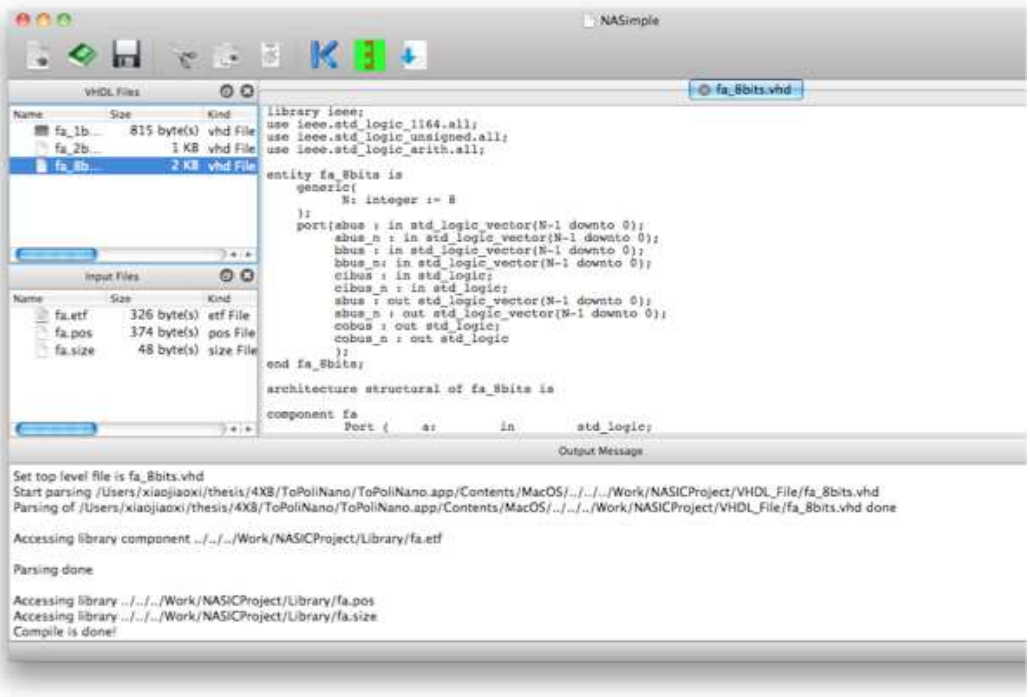



Figure 6.3. ToPoliNano GUI.

- **Simulation.** The circuit is simulated using a behavioral model. This model is based on a tristate approximation in which every magnet has only three possible states (logic '0', logic '1' and RESET). This model is validated choosing the basic logic gates from the literature or through a low level simulation of the basic logic gates, using the well established simulators OOMMF or NMAG. This approach assures the correctness of the behavior of the whole circuit, and at the same time allows the fast simulation of circuits composed by millions of magnets.

This modular structure gives to the designer a very high flexibility. For example, if new theoretical results are reached the place&route algorithm can be replaced with an optimized version to keep into account the new constraints. At the same time with this structure it is possible to extend this tool to other technologies, using as a base the existing modules, and changing only the modules that differs. Figure 6.5 shows as a proof a full adder made with Silicon Nanowire NanoPLA [50].

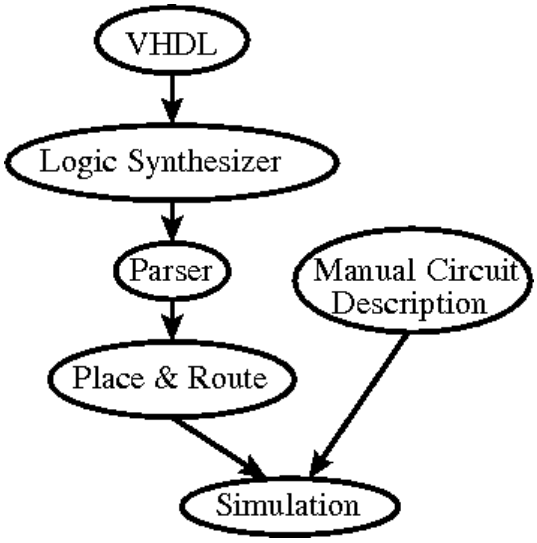



Figure 6.4. ToPoliNano design flow. *M. Vacca et al. "ToPoliNano: A synthesis and simulation tool for NML circuits", International Conference on Nanotechnology, 2012*

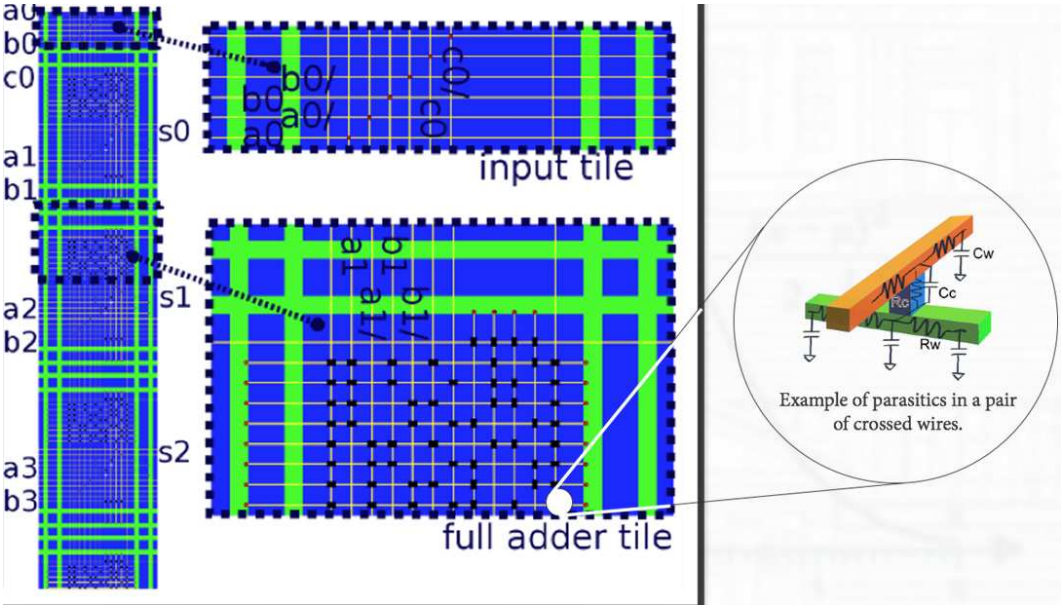



Figure 6.5. Silicon Nanowire NanoPLA full adder. *S.Frache et al. "ToPoliNano: Nanoarchitectures Design Made Real", Nanoarch, 2012*

## 6.3 Logic Synthesizer

The logic synthesizer takes as an input a circuit described using VHDL language. The description of the circuit must be as generic as possible and technology independent. In this way the same circuit can be synthesized with different technologies.

The synthesizer produces as output another VHDL file, which contains the description of the same input circuit but mapped on the chosen technology library. The synthesizer is based on SIS [51], an academic free tool for circuit synthesis, which takes a generic complex logic function and decomposes it in many small simple functions. This functions are then mapped on majority voters using an algorithm ad-hoc developed. This part of the tool is still in development, particularly the set of logic gates used for the technology mapping must be enlarged (for example considering the recently developed Magnetic And/Or gates [6]). Moreover the different part that composes the synthesizer must be fully merged and integrated into the whole software.

## 6.4 VHDL Parser

The parser is a fully developed intermediate block which translates the output circuit generated by the synthesizer into a format compatible and easy manipulable by the other parts of the tool. It accepts as an input only VHDL files with a pure structural description. Structural description means that the circuit is a pure netlist of logic gates, of any kind, which contains no others construct, like logic equations. The output is a graph which represents the circuits, described using a format compatible with the rest of the tool.

## 6.5 Manual circuits description

The tool has the option to manually describe circuits, placing components and wire manually. This is done for two reasons: to allow circuit simulations until the place&route algorithm will be finished and to design optimized versions of a specific circuits. Clearly no algorithm can reach the level of optimization that a human design can reach. Moreover, these optimized blocks can be used by the the place&route algorithm to improve the overall circuit layout. Circuit can be described in two ways: Directly writing the code in the simulator or drawing the circuit with a vector graphic program like XFIG and exporting the results on an .svg format. The .svg exported file will be automatically translated in a circuit. Figure 6.6 shows am example of circuit manually described, a 32bit ripple carry adder. On the left it is shown a full adder made with AND/OR gates while on the right there is a full adder made with majority voters [4]. This picture gives an idea of the circuit complexity that can be obtained and simulated with ToPoliNano.

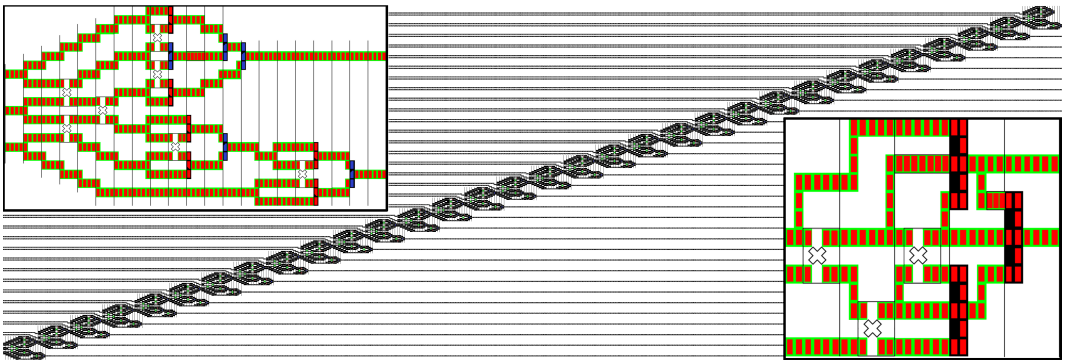


Figure 6.6. NML layout example: A 32 bit ripple carry adder. In the left detail a full adder made using AND/OR gates is shown, while in the right detail there is a full adder made with majority voters. *M. Vacca et al. "ToPoliNano: A synthesis and simulation tool for NML circuits", International Conference on Nanotechnology, 2012*

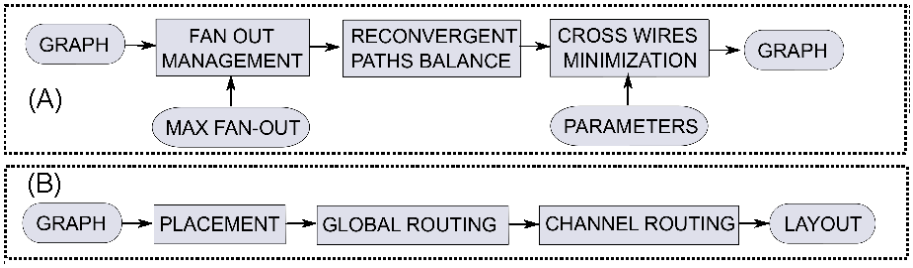



Figure 6.7. A) Graph elaboration flow diagram. C) Physical Mapping flow chart.

## 6.6 Place & Route

The place&route is the most critical block. Its final goal is to start by the graph generated by the parser, physically place every logic gates, and interconnect them routing the wires [1]. The Place&Route algorithm can be divided in two parts: Graph Elaboration, discussed in Subsection 6.6.1 used to pre-process the graph representing the circuit, and Physical Mapping, presented in Subsection 6.6.2, that finalizes the placement and routing.

### 6.6.1 Graph elaboration

The Graph Elaboration flow diagram is shown in Figure 6.7.A. The input is the graph generated by the HDL parser, which contains a structural description of the circuit mapped on the logic gates available (and, or, majority voter,

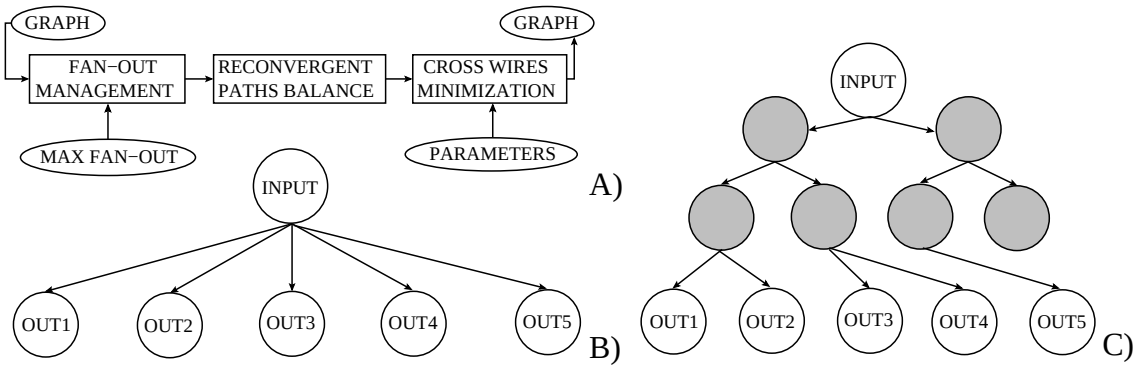



Figure 6.8. A) Graph before Fan-out Limitation is applied. B) Graph after Fan-out Limitation.

INVERTER). The circuit structure is then modified according to the intrinsic characteristics of NML logic and the clock zones layout. Three important operations are sequentially executed on the input graph: Fan-Out Management, Reconvergent Paths Balance and Wire Cross Minimization. The elaborated graph is then used as input for the PHYSICAL MAPPING.

### Fan-Out Management

Similarly to CMOS circuits also in NML technology a limitation on the fan-out of each logic gates holds. The fan-out limitation in NML is mainly related to the clock zones layout and to the physical space occupied by the wires. Particularly, with a layout of clock zones organized in parallel stripes, there is a limitation in the length of vertical magnetic wires to avoid propagation errors [39]. Moreover, every wire is made by magnets so there must be enough room to allow their physical placement. For this reasons the graph is iteratively scanned and, if nodes with more than N fan-out connections are found, additional levels and nodes are added until every node as a fan-out smaller or equal to N. The fan-out limit (N) is a parameter that must be given as input to the algorithm. This additional nodes will be physically represented by NML wires. This process can sometime generate subtrees without leafs, i.e. subtrees composed only by additional blocks created by the fan-out limitation routine that do not connect any logic gate. A specific subroutine eliminates all this dead subtrees. Figure 6.8.A shows a generic graph before the application of the fan-out limitation routine. Figure 6.8.B shows the results of the algorithm if the fan-out limit is set to 2. Clearly, the stricter is the limitation on the fan-out, the bigger is the number of additional levels.

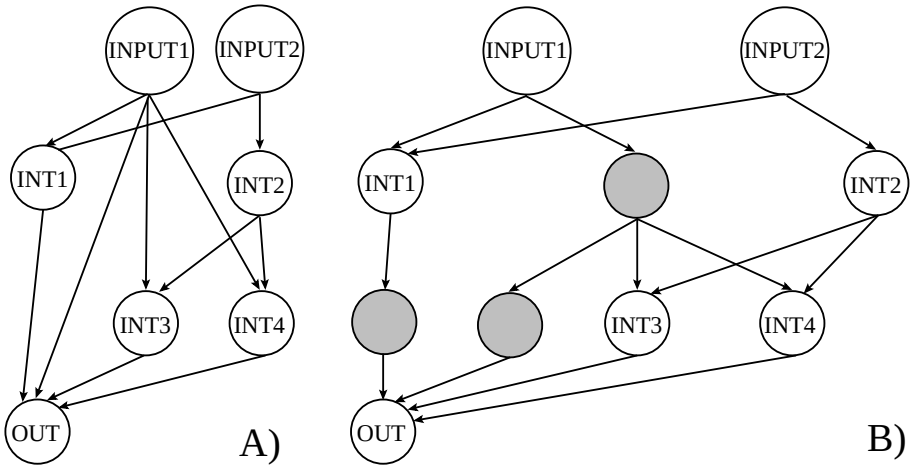



Figure 6.9. Reconvergent Paths Balance. A) Graph before leveling. B) Graph after wire block insertion and wire block sharing.

### Reconvergent Paths Balance

In a graph two paths are called *reconvergent* if they diverge from and reconverge to the same blocks. For example in Figure 6.9.A all paths start from inputs I1 and I2 and converge to the output O. This is a common situation in an electronic circuit but in NML it presents further challenges due to the intrinsic pipelined behavior. The delay in terms of clock cycles of a wire depends on its layout (problem known as "layout=timing" [2][5]). To be synchronized, signals at the input of a logic gate must have the same length and therefore the same delay. As a consequence, reconvergent paths must be balanced, i.e. they must have the same number of nodes. To obtain this result, starting from an unbalanced graph (Figure 6.9.A), intermediate nodes that physically represent NML wires are inserted in the graph to balance all the paths. The RECONVERGENT PATHS BALANCE routine can sometimes add duplicated wire nodes, i.e. wires nodes that have the same father but different children. A subroutine of WIRE SHARING merges these nodes together reducing the complexity and optimizing the graph. The results of the RECONVERGENT PATHS BALANCE algorithm are shown in Figure 6.9.B. Every path is balanced granting perfect signal synchronization.

### Wire Cross Minimization

One of the most characteristic features of NML and QCA technology (from which NML is derived as a particular implementation) is that both logic gates and interconnections are on the same plane. Up to now there are no experimental evidences that it will be possible to route wires on different layers. As a consequence in this

technology a particular block is available, the cross-wire, that allows the crossing of two wires on the same plane. Even though it enables the routing of signals on only one layer, the circuit layout must be optimized to reduce the number of cross-wires required, therefore reducing the wasted area. Different techniques can be used for minimizing wire crosses. Previous works [52][53] take advantage of the clocking constraints to propose simple methods for cell placement, belonging both to the analytical and stochastic families. To the analytical set belongs the *Barycenter method* and the *Fan-out Tolerance Duplication*, while *Simulated Annealing* is a stochastic method. Not only we implemented all these algorithms, but we further enriched the range of possibilities available to the user with a partitioning algorithm exploiting and improving *Kernighan-Lin* heuristic.

### Wire Cross Minimization - Barycenter

The *Barycenter* method is a very simple technique that changes the position of each node on the graph trying to place every node directly above the nodes to which it is connected. This algorithm is in this work coupled with a fan-out duplication technique, where nodes are duplicated to reduce the fan-out of each node, improving the performance of the *Barycenter* method.

The BARYCENTER algorithm explores each rows of the graph two by two from inputs to outputs. For each couple of rows analyzed one is kept frozen while nodes in the other row are changed in position to reduce the number of crosswires. At each node is assigned a weight based on the average column of its fan-outs or fan-ins. The aim of the algorithm is to place nodes directly above their fan-in or fan-out nodes reducing the average distance between parents and children and eliminating unnecessary wire crossings. After two rows are optimized the subsequent set of rows is selected. This set is composed by the bottom row of the previous considered pair and a new row. So for each set of rows optimized the frozen row is alternatively the bottom row and the upper row. This is done for a simple reason. Let's suppose that nodes of a layer are moved according to their weight. The algorithm then jumps to the next pair of rows. If the first row of this set is moved, the optimization done in the previous step is no more effective because this row was the bottom frozen row in the previous algorithm step. So in this case the upper row is kept frozen and the nodes of the bottom rows are swapped. Figure 6.10.A shows an unoptimized situation, while Figure 6.10.B shows the results of the BARYCENTER algorithm. This algorithm is quite simple and fast but leads to an unoptimized result. This is due because there can be situation in which multiple solutions satisfy the requisites of the algorithm. This solutions have however a different number of crosswires, so the efficiency of the algorithm heavily depends on the policy chosen to solve this conflicting situations.

The probability to have a wire crossing heavily depends on the length of the

wire: the higher the length the higher the probability is. In a k-layered bipartite directed acyclic graph (KLBDAG), which are the kind of graphs that we have to handle (Figure 6.10), the length of a wire can be estimated by the number of column traversed by the connection edge. However if the fan-out of a node is equal to 1 the nodes position can be switched placing the two nodes in the same column, reducing the wire length and the number of crosswires. The FAN-OUT DUPLICATION algorithm is an integration to the BARYCENTER algorithm. A threshold can be set as a parameter. This threshold represents the maximum number of column that a graph edge can traverse. All nodes with fan-out higher than 1 where destination nodes are apart farther than the threshold, are duplicated. BARYCENTER algorithm is then applied to minimize wire crossing.

| Thres. | %Blocks increase | %Crosses reduction |
|---|---|---|
| 0 | +780% | -100% |
| 1 | +300% | -85% |
| 2 | +84% | -55% |

Table 6.1. Results of Fan-out Tolerance Duplication for different thresholds [1]

Table 6.1 shows the results on a random generated graph. The weakness of this technique is due to the fact that when a node is duplicated all its input nodes must be duplicated. As can be seen from Table 6.1 with a threshold of 0 all the crosswires are eliminated but at the cost of am explosion of the circuit area. Increasing the

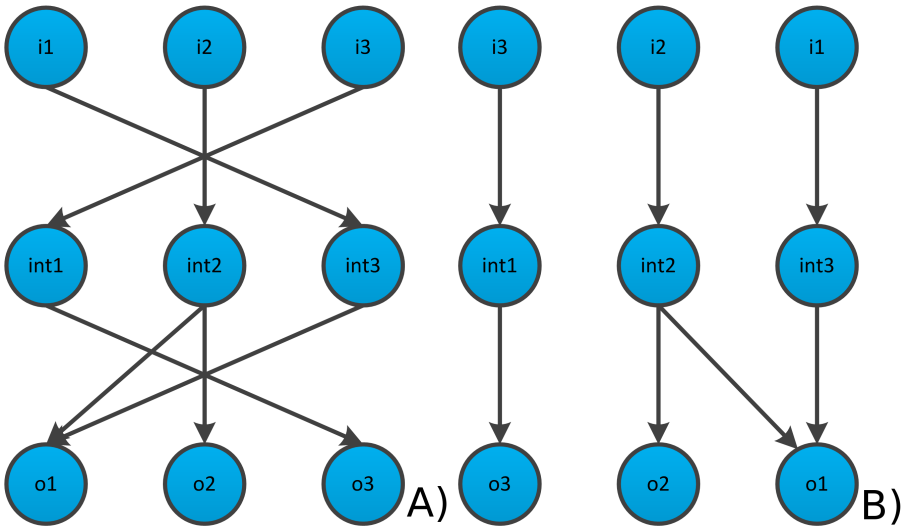



Figure 6.10. Graph before (A) and after (B) Barycenter application [1]

threshold value leads to a reasonable crosswires reduction with a relative small increase in the graph size.

### Wire Cross Minimization - Kernighan-Lin

One of the most commonly used algorithm in the class of partition based placements is the so called Kernighan-Lin. This algorithm heuristically divides the graph into sub-regions, trying to minimize the cut, i.e. the number of edges that connect one sub-region from the others. The pseudocode shown in listing 1 explain how the algorithm minimize the cut given two sets of nodes.

$A^0$ and $B^0$ are two subsets of nodes that contains the same number of elements. For every nodes of partition $A$ the gain $D_a$ is evaluated (rows 6-8, listing 1). The gain of a node is given by

$$D_a = E_a - I_a \tag{6.2}$$

where $E_a$ is the sum of the weights of outgoing arches, i.e. those starting in a partition and ending in the complementary one ("External cost"), while $I_a$ is the weighted summation of all edges starting and ending in the same partition ("Internal Cost"). The same process is repeated for all the nodes of partition $B$.

Now that the gain is determined, for all possible couples of nodes, one for each partition, the total gain is evaluated

$$\Delta = D_a + D_b - 2\gamma_{ab} \quad a \in A^{m-1} \quad b \in B^{m-1} \tag{6.3}$$

The total gain is obtained adding the gain of each individual node $D_a$ and $D_b$ and subtracting $2\gamma_{ab} \quad a \in A^{m-1} \quad b \in B^{m-1}$, which is the total sum of the weights of the edges that connect the two nodes. Once the maximum value of $\Delta$ is individuated the correspondent couple of nodes $a$,$b$ is swapped and locked. The gain of each node is now reevaluated. This is repeated until there are unlocked nodes. The value of $G$ identify the sum of the total gains of each couple of nodes. The aim of the algorithm is to obtain a value of $G$ lower than 0, so this entire procedure is repeated until this results is reached. After the cut is minimized each partition is divided in two subpartition and the whole process is iteratively repeated until each partition is composed by only one nodes and no further divisions are possible.

While in CMOS every node of the graph can be swapped in NML there is not this freedom. In NML the graph is divided in rows (Figure 6.11), where every row represents a clock zone and only the nodes that are part of the same row can be swapped. Figure 6.11 shows an example of how the graph is partitioned. The maximum width of the graph, in terms of number of nodes, is calculated. Nodes are equally distributed along the rows. The graph is then cut in two parts, with the cut that is placed exactly in the middle of the rows. Thanks to the fact that nodes are equally distributed along the rows this two subsets of rows contains exactly the same number of nodes.

**Algorithm 1** KL pseudocode

```
1: procedure KL(V)
2:     initialize(A^0,B^0)
3:     m ← 1
4:     repeat
5:         for all a ∈ A^{m−1} do
6:             compute D_a
7:         end for
8:
9:         for all b ∈ B^{m−1} do
10:            compute D_b
11:        end for
12:
13:        for i ∈ [1,n] do
14:
15:            find unlock vertices a_i ∈ A^{m−1}, b_i ∈ B^{m−1}
16:            such that Δ_i = D_{a_i} + D_{b_i} − 2γ_{a_i b_i}
17:            is maximal
18:
19:            lock a_i and b_i
20:            for all unlocked x ∈ A^{m−1} do
21:                D_x ← D_x + 2γ_{x a_i} − 2γ_{x b_i}
22:            end for
23:            for all unlocked y ∈ B^{m−1} do
24:                D_y ← D_y − 2γ_{y a_i} + 2γ_{y b_i}
25:            end for
26:        end for
27:
28:        find k such that Σ_{i=1}^{k} Δ_i is maximal
29:        G ← Σ_{i=1}^{k} Δ_i
30:
31:        if G > 0 then
32:            X^m ← a_1,a_2,...,a_k
33:            Y^m ← b_1,b_2,...,b_k
34:            A^m ← (A^{m−1} \ X^m) ∪ Y^m
35:            B^m ← (B^{m−1} \ Y^m) ∪ X^m
36:            Unlock all vertices in A^m and B^m
37:            m ← m + 1
38:        end if
39:
40:    until G ≥ 0
41: end procedure
```

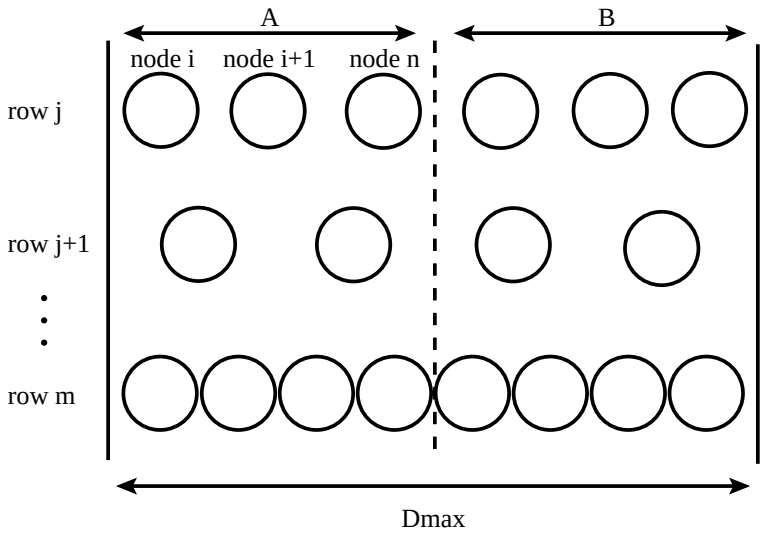



Figure 6.11. KL algorithm applied to NML circuits.

The KL modified algorithm is shown in listing 2. The modification are built around the calculation of total gain of each couple of nodes. While in listing 1 (row 13) every couple of nodes inside the circuit are considered, in listing 2 (rows 13), every row is considered alone and all the nodes that are part of that row are considered. In CMOS are normally necessary four iteration of the algorithm to reach the optimum solution. However in NML the graph structure is slightly different, so the optimum solution is found after only one iteration. An example of how the KL algorithm works is shown in Figure 6.12. As it is possible to see when the swap gain reach the minimum value new partition are created. The process is repeated until the Gain History reaches the minimum value.

### Wire Cross Minimization - Simulated Annealing

Simulated annealing is a stochastic technique that iteratively swap the position of nodes inside the graph trying to find the global minimum through consecutive solutions. The pseudocode is reported in listing 3.

The algorithm needs 3 data as input, the graph nodes $V$, the minimum temperature that must be reached and the number of iteration that are necessary to obtain the final state. The parameters used determine the efficiency of the simulated annealing algorithm, and this parameters must be chosen according to experience. In the TimberWolf [54] placer the final temperature is 0.1 while the number of iteration is a number that depends on the circuit complexity. We have chosen to make a maximum number of 500 iterations for a circuit of 30000 elements, because

**Algorithm 2** KL pseudocode modified.

```
1: procedure KL(V)
2:     initialize(A^0,B^0)
3:     m ← 1
4:     repeat
5:         for all a ∈ A^{m−1} do
6:             compute D_a
7:         end for
8:
9:         for all b ∈ B^{m−1} do
10:            compute D_b
11:        end for
12:
13:        for j ∈ [1,m] do
14:            for i ∈ [1,n] do
15:
16:                find unlock vertices a_i ∈ A^{m−1}, b_i ∈ B^{m−1}
17:                such that Δ_i = D_{a_i} + D_{b_i} − 2γ_{a_i b_i}
18:                is maximal
19:
20:                lock a_i and b_i
21:                for all unlocked x ∈ A^{m−1} do
22:                    D_x ← D_x + 2γ_{x a_i} − 2γ_{x b_i}
23:                end for
24:                for all unlocked y ∈ B^{m−1} do
25:                    D_y ← D_y − 2γ_{y a_i} + 2γ_{y b_i}
26:                end for
27:            end for
28:        end for
29:
30:        find k such that Σ_{i=1}^{k} Δ_i is maximal
31:        G ← Σ_{i=1}^{k} Δ_i
32:
33:        if G > 0 then
34:            X^m ← a_1,a_2,...,a_k
35:            Y^m ← b_1,b_2,...,b_k
36:            A^m ← (A^{m−1} \ X^m) ∪ Y^m
37:            B^m ← (B^{m−1} \ Y^m) ∪ X^m
38:            Unlock all vertices in A^m and B^m
39:            m ← m + 1
40:        end if
41:
42:    until G ≥ 0
43: end procedure
```

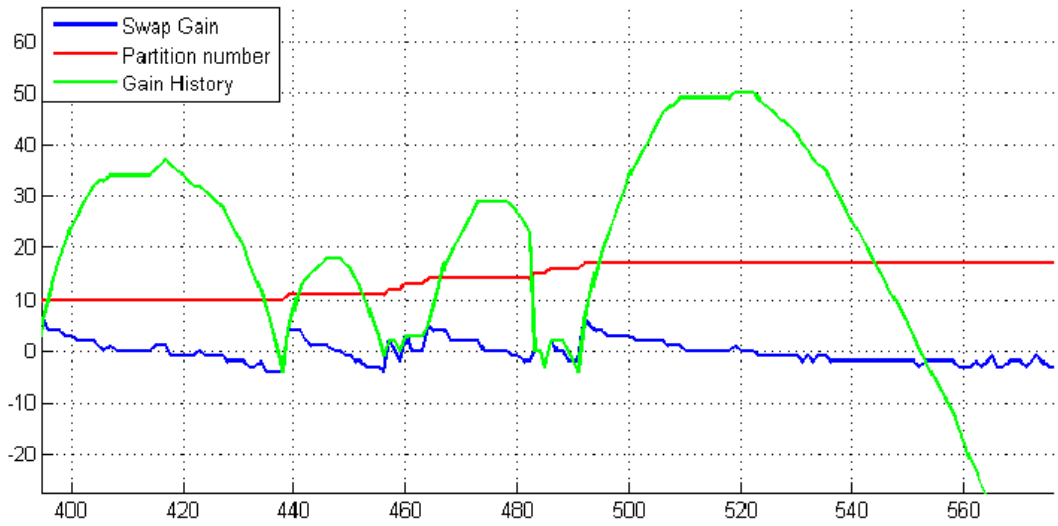



Figure 6.12. Gain history for the entire set of partitions [1].

according to our experience this is a number that allows us to reach a good solution. At the beginning of the algorithm the temperature is set to its initial value value. In the TimberWolf placer the initial value is set to 4000000. However we apply the simulated annealing only after the application of the barycenter technique. As a consequence when the simulated annealing is applied the number of wire crossing is already greatly reduced so it is possible to choose a much smaller value of temperature (5000 in our case) speeding up the whole algorithm.

The algorithm itself is quite simple. During each iteration a random level is chosen (rows 8, listing 3). In the chosen level two random nodes are selected (rows 9 and 10, listing 3). The two nodes are swapped and the $\Delta_c$ is evaluated. The $\Delta_c$ is calculated subtracting the number of wire crossing with the nodes swapped and the number of wire crossing before the node swap. If $\Delta_c < 0$ the graph is updated, otherwise a random number between 0 and 1 is calculated. If this number is lower than a value obtained by a probabilistic function than the graph is updated otherwise the nodes are swapped back in the original position. This probabilistic function depends on the value of $\Delta_c$ and $T$. Thanks to this trick, simulated annealing avoids getting stacked at the local optimum by occasionally accepting moves that result in a cost increase. These moves are accepted according to a probability that depends on the temperature of the algorithm T. Typically higher cost choices are accepted for higher temperatures.

After the set number of iteration the value of temperature is updated and the whole process is update. The temperature is updated according to the profile store in the function tempScheduler(). Particularly we have implemented two profiles:

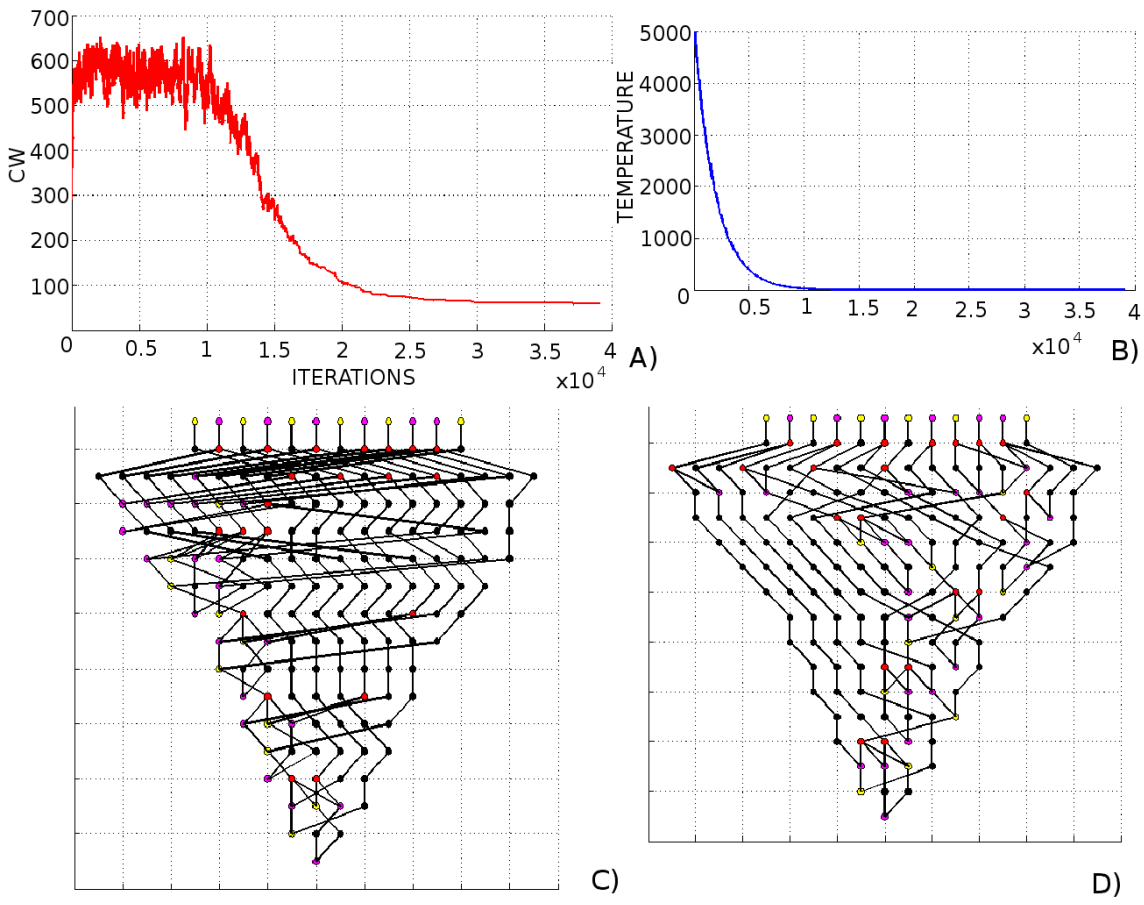



Figure 6.13. A) B) Simulated Annealing applied to a 6 bit RCA. C) D) Graph processing through SA, PT=29.8 s [1].

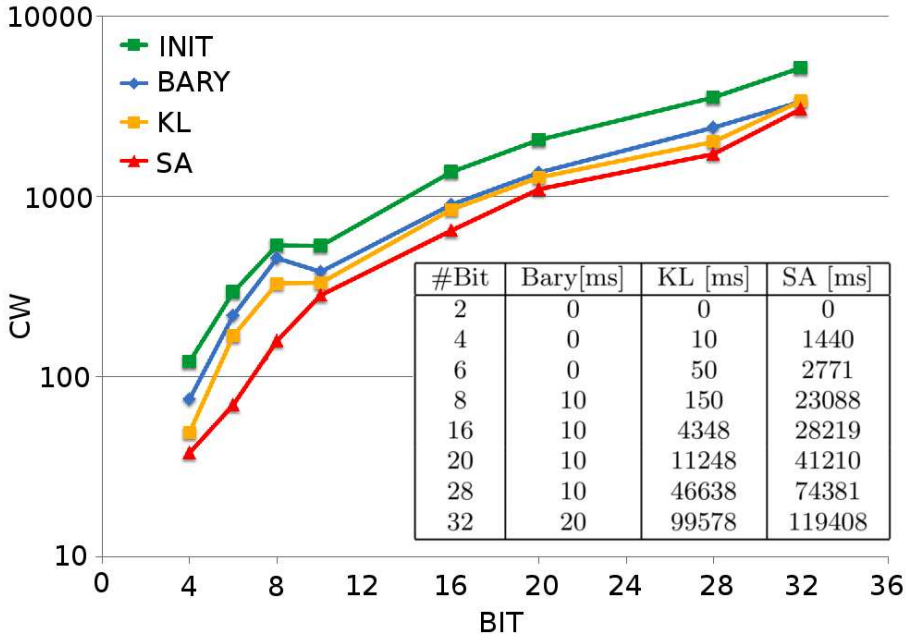
An decreasing exponential function and a function similar to the one used in the TimberWolf, which has a lower slope at the end and at the beginning but a higher slope in the middle. Figure 6.13.A shows how the number of crosswires change with the step iterations. At the beginning there is a huge oscillation in the number of wire crossing, but with the successive iterations it decreases to its minimum value. Figure 6.13.B shows the temperature profile used, while Figure 6.13.C shows a graph before the application of simulated annealing and Figure 6.13.C shows the same graph after the application of simulated annealing. While simulated annealing can lead to very good results, its a stochastic method that heavily relies on the parameters used, on the function used to generate random numbers and generally requires a huge time to converge.

**Algorithm 3** SA pseudocode

```
1: procedure SA(V, T_min, numIter)
2:     T ← T_0
3:     P=Place(V)
4:
5:     while T > T_min do
6:
7:         while it¡numIter do
8:             l ←random(1,DEPTH)
9:             a ←random(1,nodeList(l))
10:            b ←random(1,nodeList(l))
11:
12:            while b!=a do
13:                b ←random(1,nodeList(l))
14:            end while
15:
16:            newP ← swap(a,b)
17:            Δcost = cost(newP)-cost(P)
18:
19:            if Δcost¡0 then
20:                P=newP
21:            else
22:                r=random(0,1)
23:                if r < e^(−Δc/T) then
24:                    P = newP
25:                end if
26:            end if
27:
28:        end while
29:
30:        T ← tempScheduler()
31:
32:    end while
33:
34: end procedure
```

### Comparison among crosswires minimization techniques

Figure 6.14 shows the comparison of crosswires minimization techniques we implemented. The test circuit is a N bit ripple carry adder, with N varying from 4 to 32 bits. Some important conclusions can be derived from Figure 6.14. The number of crosswires increases with the growing number of bits, however all the techniques allow a consistent reduction of the number of crosswires. In case of a 32 bit adder the number of crosswires is reduced from 5000 to about 3000. The Barycenter algorithm is the simplest but it offers the worst performance. The Kernighan-Lin modified algorithm offers better performance but not so good as the Simulated Annealing. The advantage of Simulated Annealing is greater with a small number of bits but it is reduced for an high number of bits, where the results of all algorithm tend to saturate.




| #Bit | Bary[ms] | KL [ms] | SA [ms] |
|---|---|---|---|
| 2 | 0 | 0 | 0 |
| 4 | 0 | 10 | 1440 |
| 6 | 0 | 50 | 2771 |
| 8 | 10 | 150 | 23088 |
| 16 | 10 | 4348 | 28219 |
| 20 | 10 | 11248 | 41210 |
| 28 | 10 | 46638 | 74381 |
| 32 | 20 | 99578 | 119408 |

Figure 6.14. Wire cross reduction comparison of different algorithms. A multi bit adder is used as benchmark. Inset with table: Execution time for wire cross minimization algorithms applied to a variable bit number Ripple Carry Adder [1].

However if we look at the time requested to obtain these results shown in the figure 6.14 inset with tabulated data: Barycenter method (Bary), Kernigan-Lin modified method (KL), Simulated Annealing method (SA). We observe that the relative small improvement offered by Kernighan Lin and Simulated Annealing algorithms has an high cost in terms of computational time. While the application

of BARYCENTER method to a the 32 bit case takes only 32 ms, the other techniques require 1-2 minutes. It is clear from this data that KERNIGHAN LIN and SIMULATED ANNEALING algorithm must be used only on relative small circuits and only when a very high level of optimization is requested.

### 6.6.2 Physical Mapping

In the PHYSICAL MAPPING process the graph is translated into the circuit layout. The general flow chart is shown in Figure 6.7.B. After every node is mapped to its correspondent logic gate, it is PLACED in the circuit. A GLOBAL ROUTING phase follows where an approximated routing is performed and the position of each gate is changed trying to obtain the minimum area solution. When the position of each logic gate is defined a DETAILED ROUTING among the blocks flows.

#### Placement

As a first step every node is mapped to its correspondent logic gate. The placement of these logic gates follows the structure of the graph. As shown in Figure 6.15.A, every node of the graph is placed, row by row. Each node corresponds to a logic gate with a different area. Logic gates are placed without any optimization (Figure 6.15.B), with the each base side aligned at the beginning of the row. After this phase it is possible to evaluate the minimum area requested for the placement of the circuit. Finally the position of each gate is shifted using a simple Barycenter approach (Figure 6.15.C). The final position of each gate will be decided at the end of the GLOBAL ROUTING phase (see later). It is important to underline that the placement of the circuit relies

on the clock structure described in [2], where clock zones are organized in parallel strips. Thought other organizations are theoretically possible, this layout is chosen

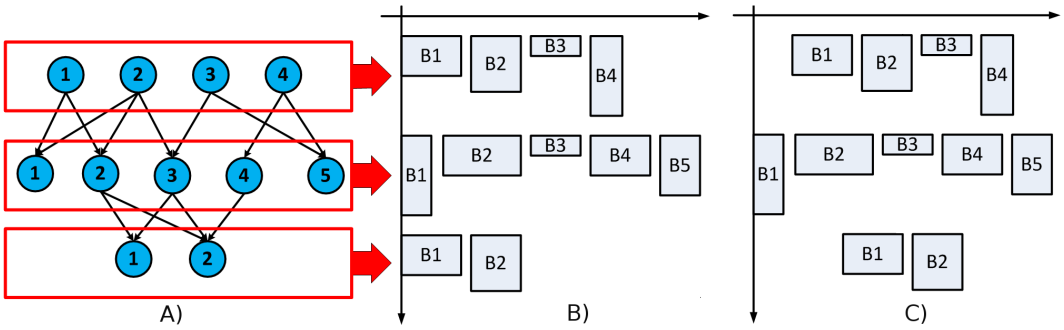



Figure 6.15. A-B) Seed row placement for maximum width evaluation. C) Barycentered placement [1].

here because it is the only one currently experimentally demonstrated and because it is well suited for combinational circuits with a dataflow structure.

### Global Routing

To obtain the definitive positions of logic gates a Global Routing phase is required. The aim of this part of the algorithm is to find the optimal shift of the position of each logic gate. The reason is to reduce the length of the interconnection wires, obtaining therefore the global minimum area of the circuit. The flow diagram is shown in Figure 6.16.A. It is an iterative process where for each couple of rows i) logic gates are shifted, ii) interconnection wires are routed and iii) the interconnections area is evaluated. Every wire is based on magnets that must have a minimum separation between them, so the area occupied by the wires can be easily evaluated. The structure of the circuit can be divided in rows, corresponding to the graph nodes that represent logic gates, separated by channels dedicated to interconnections. The aim of the Global Routing phase is to obtain the minimum width for each routing channel. The results obtained at the end of this phase are two: 1) The final position of each logic gate and 2) the final position of the pin, the input and output points in the routing channel. Figure 6.16.B shows an example of two rows before the Global Routing phase, while Figure 6.16.C shows the situation after the minimum is reached. Gates position is shifted and the length of the routing channel is greatly reduced. From Figure 6.16.C the input and output points of the routing channel can be observed.

### Channel Routing

Now that the final position of each gate is defined, wires can be routed and the final circuit layout obtained. This part of the algorithm is called Channel Routing. It takes as input a channel with input and output signal positions fixed (Figure 6.17.A) and places interconnections wires. In CMOS technology routing is normally performed on Manhattan Grids, with interconnections made by horizontal and vertical segments perpendicular among them. This solution is not well suited for NML technology. The reason lies in the particular clock zones layout [2] and on the limited number of elements that can be cascaded avoiding errors in the signal propagation [39]. The consequences are that signals can propagate without problems in the direction perpendicular to clock zones strips, but not so in the other direction. The propagation in this second direction follows a stair-like pattern. Signals, then, can only move in oblique (up or down). For this reason a Manhattan Grids approach cannot be used. The approach that we have chosen is called mini-swap [55], and it is shown in Figure 6.17.B. Interconnection wires are routed in a oblique way. When two nets cross, then they are physically mapped with a crosswire block (Figure

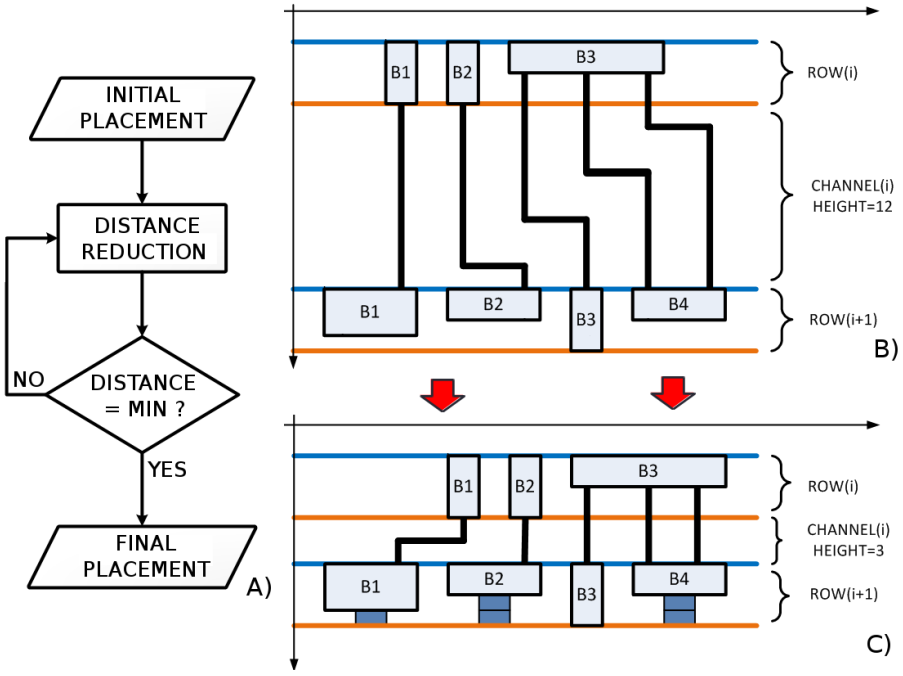



Figure 6.16. A) Global Routing flow diagram. B) Unoptimized placement. C) Optimized placement. [1]

6.17.C). Finally the oblique interconnections are mapped in the real circuit using magnets. Figure 6.17 shows a detail of the final resulting circuit. The "stair-like" signal propagation, which is typical of this technology, is evident. The maximum number of magnets that can be cascaded in one direction or in the other direction is a parameter that can be set by the user, and it will affect the final layout and area of the circuit.

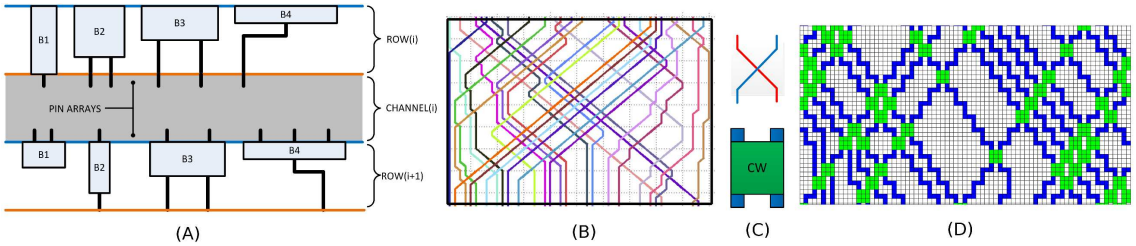



Figure 6.17. A) Pins for channel definition. B) Mini Swap model for channel routing. C) Crosswire mapping. D) Physical mapping of interconnections. [1]

### Results

Figure 6.18 shows an example of circuit layout (rotated by 90 degrees) obtained at the end of the whole algorithm. It is a 6 bit ripple carry adder with a total area of 30x5 $um^2$, while the size of each magnet is 60x90 $nm^2$.

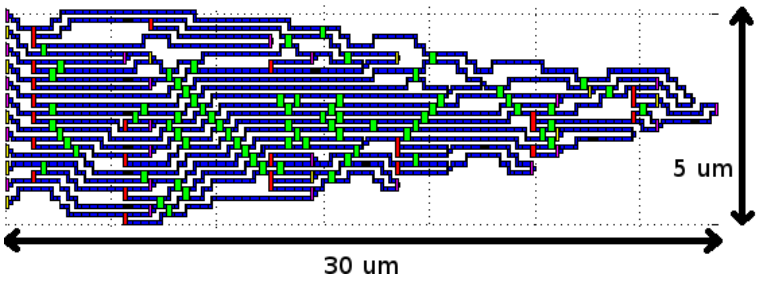



Figure 6.18. Layout of a 6 bit Ripple Carry Adder [1].

Table 6.2 shows the results obtained on some of the ISCAS85 benchmark circuits. The generation of the layout requires a time (PT) between few milliseconds and few minutes depending on the circuit complexity (# cells). The circuit area (CA) is of few $um^2$ for the c17 which is made by 7 logic gates, while the most complex (c6288) has an area of nearly $1mm^2$, as it is composed by nearly 350000 logic gates. it is interesting to point out from the last column of Table

6.2 that there is a lot of wasted area, since the area occupied by logic gates (% OCC) is at maximum 30% of the total circuit area. This is natural due to the single layer used to place both gates and interconnects.

To compare NML with CMOS same data on ISCAS benchmarks were generated and not reported for the sake of brevity. More interesting is here to focus on a single RCA if increasing number of bit. Figure 6.19.A shows area and power dissipation of RCA for both technologies. Data for CMOS are obtained through synthesis on a liberty library file. NML data on area are a result of this work, while power analysis uses the resulting number of magnets and power dissipation models in [2]. The layout

| circuit | PT [ms] | #cells | CA [$mm^2$] | %OCC |
|---|---|---|---|---|
| c17 | 88 | 7 | 0.0000035 | 30 |
| c880 | 5919 | 5753 | 0.00641 | 32 |
| c1908 | 54724 | 6941 | 0.0107 | 32 |
| c2670 | 125818 | 19350 | 0.0292 | 29 |
| c3540 | 134712 | 42822 | 0.0768 | 22 |
| c5315 | 608320 | 192159 | 0.282 | 32 |
| c6288 | 661931 | 349083 | 0.94 | 25 |
| c7552 | 465690 | 185879 | 0.261 | 33 |

Table 6.2. Results of ISCAS85 samples [1].

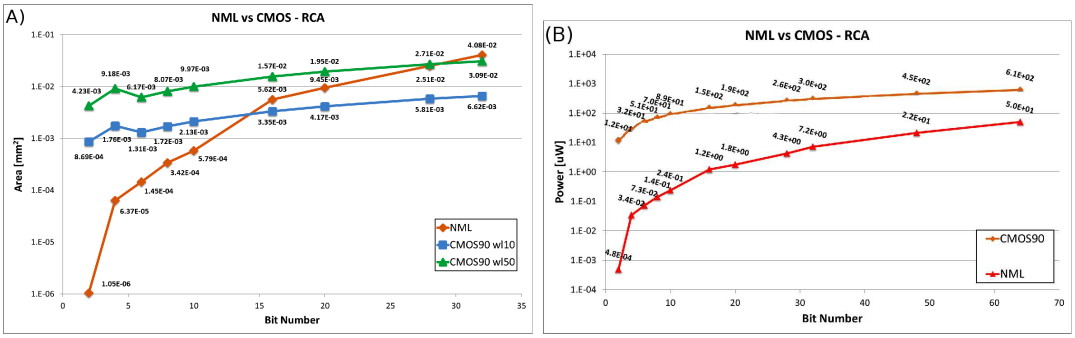



Figure 6.19. A) Comparison for RCA between NML and CMOS 90 nm in terms of area (two wireload models). B) Comparison for RCA between NML and CMOS 90 nm in terms of power dissipation. [1]

generator is very effective with small word widths: for RCA the resulting area is smaller up to 14 bit parallelism when compared to CMOS. In other words, NML area occupation converge to CMOS only for circuits of medium complexity (from 1000 to 10,000 cells). Figure 6.19.B show results on power consumption. Power savings range from 1 to 5 orders of magnitude compared with CMOS 90 nm technological nodes. If NML cannot be competitive in terms of frequency (around 100-200MHz are the expected frequency ranges), area is competitive up to medium size circuits due to the single layer available. However, a huge advantage in terms if power consumption is expected. This suggests it is worth inspecting the future evolutions of NML and prompts to move toward a multilayer organization of interconnects.

## 6.7 Simulator

The idea behind the entire tool is to quickly describe and simulate complex circuits. As a consequence the simulation engine cannot be based on the LLG equation, which accurately describes the magnetodynamics. The simulation engine uses therefore a simplified model based on low level simulations and experimental results, allowing to obtain relatively accurate results with a fast simulation.

### 6.7.1 Swich model

The simulation engine considers every magnet like a tristate device, where the possible states are logic '0', logic '1' and RESET. This is similar to what happens in digital simulators for CMOS, like Modelsim [38]. The aim of this approach is clearly to maximize simulator performance.

The basic idea behind the switch model is shown in Figure 6.20. Magnets are

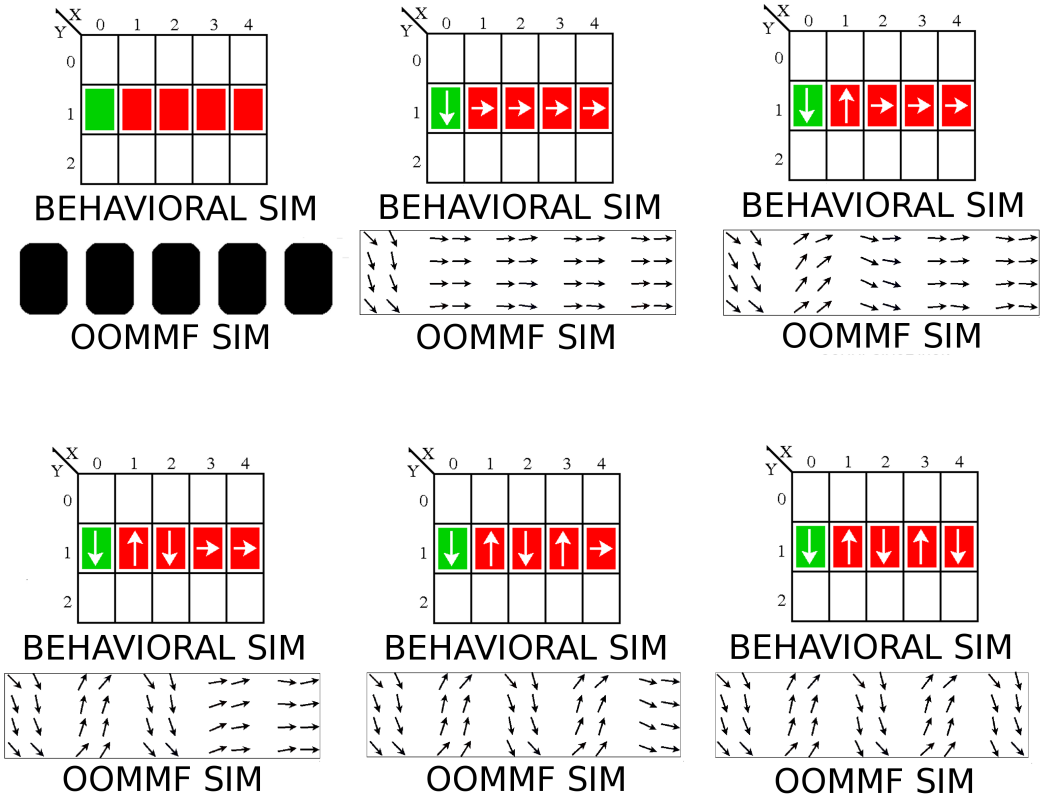



Figure 6.20. Topolinano switch model. *M.Vacca et al."ToPoliNano: A synthesis and simulation tool for NML circuits", International Conference on Nanotechnology, 2012*

forced in the RESET state. When the field is removed magnets start to switch from the input element with a "Domino-like" effect. Magnets assume therefore the state "Up" or "Down" depending on the neighbor elements. Validation of this model is done or using low level simulations or from experimental evidences. Figure 6.20 compares the behavior of ToPoliNano simulation engine with the results obtained from a LLG-based simulator like OOMMF. While with this model the dynamic of the circuit is lost, the final state of each magnets is correct. The reason behind the choice to analyze both simulations and experimental results in the development of the simulation engine lies in the fact that not always micromagnetic simulations and experimental results agrees. An example is the majority voter used for the full adder in the right detail of Figure 6.6. Simulations shows that it not always work, but experimental evidences [4] show that this structure work.

### 6.7.2 Clock generation

A waveform generator is used to build the clock signals (Figure 6.21.B) necessary for the circuit. The clocking scheme selected is the 3-phase overlapped clock described in Chapter 1. In this clock system 6 different states can be recognized and they are shown in Figure 6.21.B. In states 1, 3 and 5 (Figure 6.21.B) only one clock signal is high, and this means that respectively clock zones 1, 2 and 3 are in the reset state. During states 2, 4 and 6 (Figure 6.21.B) clock signals are overlapped and two types of clock zones are in the reset state simultaneously. As a consequence a finite state machine is used to represent the time evolution in the simulator (Figure 6.21.A). At the power on the FSM is in the state 0 and then it goes in the state 1, 3 or 5 depending on which is the first zone that must be reset at the beginning. Then the FSM cycles from states 1 to 6, because of the circuit periodic behavior. During each state of the FSM the 3 types of clock zones of the circuit are forced into a different state according to Figure 6.21.B. For example in the state 1, clock zones of type 1 are in the RESET state, while clock zones of type 2 are in the HOLD state and clock zones of type 3 are in the SWITCH state. The future state (Sf) is calculated starting from the present state (Sp) using the formula $S_f = (S_p + 1) mod 6$.

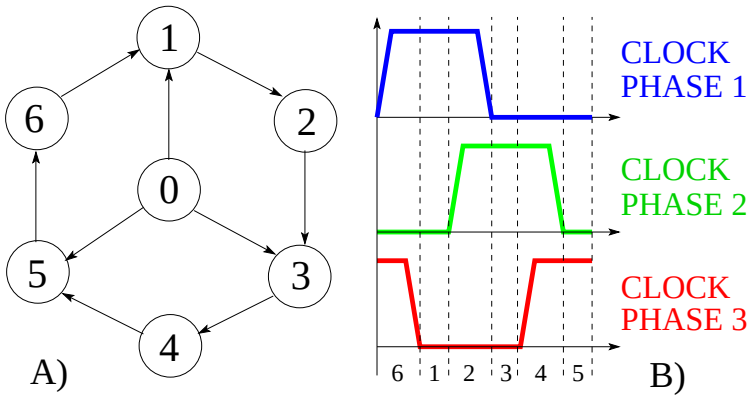



Figure 6.21. A) Finite state machine used for the state calculation B) Three phase overlapped clock and the 6 states that characterize it.

### 6.7.3 Input generation and simulation data structure

The simulation engine requires two different kind of input before being able to perform the calculations: a description of the circuit under test and all the stimuli that must be applied to the circuit. The circuit description is obtained by the place&route part of the tool, or it is manually described by the users, in order to get a full in-memory representation of the actual circuit. But what is most important

is that the data structure adopted is good for representation of the circuit but it is not optimized for simulation, so it needs further elaboration.

As a consequence the original graph (obtained from the place&route or manually designed) gets explored and a new structure, optimized for simulation purposes, gets dynamically instantiated, as a function of the hardware configuration of the host machine. This enables the software to handle even complex designs with good efficiency. The simulation uses a dynamically instantiated matrix (Figure 6.22), with dynamic entries that depend on the kind of simulation performed. The whole layout is considered like a unique collection of magnets without any distinction between magnets that are part of a logic gate and other magnets. Each element of the matrix corresponds, in general, to an area of the circuit, not directly to one of the nanomagnets. Each nanomagnet can cover more than one matrix entry. Therefore, it is possible to implement algorithms that trade memory and timing for accuracy or vice-versa. It is not necessary to take the circuit as a whole: it can be partitioned. This can be particularly useful on machines with standard hardware configurations, such as Desktop PCs.

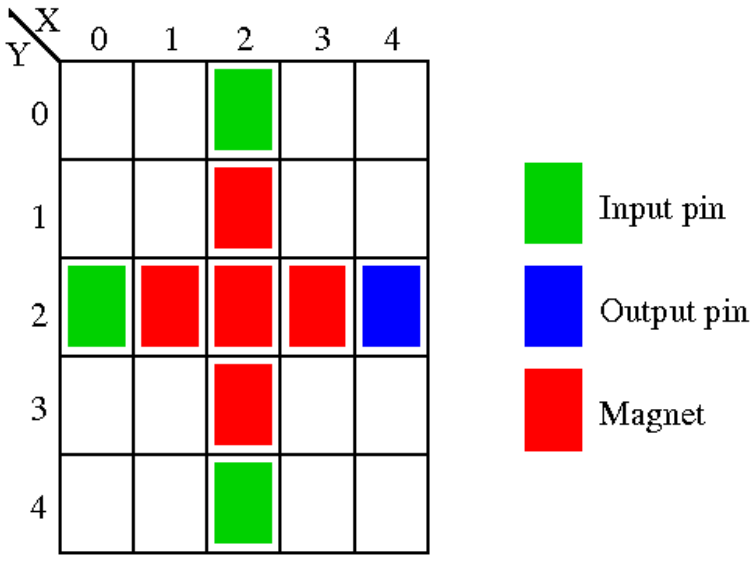



Figure 6.22. ToPoliNano simulation matrix.

Inputs are provided to the simulation engine through a vector waveform file. This file contain all the parameters necessary for the simulation. This parameters include time unit, simulation start time, simulation stop time, signal rise time, signal fall time, input signals names, input signals types and input logic levels. A transition list defines the logic levels of the signals and their duration. The time samples that must be used in the simulation are obtained from the stimuli description contained

in the waveform file, which is parsed and all the required information are placed inside a in-memory structure, to be available during the simulation so that they can be fed to the circuit. The number of time steps can be huge, depending on the time resolution and the total simulation time, so they cannot be kept in-memory and are therefore saved on the disk and recalled only when they are necessary.

Every magnet in the circuit can be selected as input/output pin, granting a very high flexibility to the designers, which can place as many inputs/outputs as necessary anywhere in the circuit.

### 6.7.4 The simulation controller

It represents the part of the program which handles all the task related to the simulation. First of all it loads inside the memory the circuit layout. Secondly the simulation matrix must be initialized, starting from the circuit layout and a certain number of parameters (i.e. the floor plan size, the width and the height of the magnetic elements grid, etc...). Only at this point the simulation controller can run the simulation. It gets the time samples at simulation runtime, just when they are needed and just after they have been concurrently created. After this it must calculate the magnetization of all magnets in the circuit and, eventually, it must generate the output value, saving the results in graphical form and by logging data onto disk.

### 6.7.5 Simulation algorithm

The simulation algorithm is quite simple. During every FSM state the value of the magnets of each clock zone of the whole circuit is calculated. If clock zones are in the HOLD state magnets are left untouched, while magnets of the clock zones in the RESET state are reset. The magnets value inside every clock zones in the SWITCH state is then calculated.

### 6.7.6 Matrix exploration

For each clock zone in the SWITCH state the first step is to identify all the magnets of a clock zone calculating clock zones borders. After the borders identification the algorithm starts the evaluation of the magnet state. The algorithm explores the matrix considering all the columns, one by one. The coordinate x is incremented linearly from $x_{west}$ to $x_{east}$ in case of forward propagation and from $x_{east}$ to $x_{west}$ in case of backward propagation. For each column every cell is considered starting from the northern border. If the cell is void, or if there is a magnet with all the neighbors in the reset state, the state of the cell is undefined and the value of the y coordinate is incremented. If there is a magnet and at least one of its neighbors is

in the stable state then the value of magnet is calculated. When the bottom of the matrix is reached, the calculation of magnets state is performed again starting from the bottom until the top is reached. In this way, magnets in which the state was undefined in the previous passage are calculated. Then the value of the x coordinate is incremented and the process is repeated until the east or west border is reached.

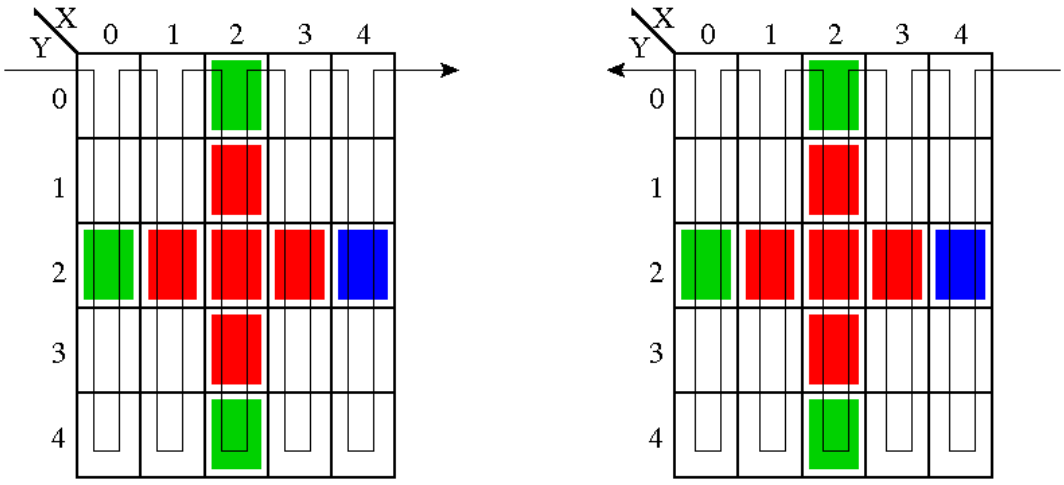



Figure 6.23. Details on matrix exploration.

Figure 6.23 shows details on how the matrix is explored. This approach is used to minimize the use of "if...then...else" constructs, that have a huge impact on performances. However exploring the matrix in this way, when it is the moment to evaluate the magnetization of a magnet not always all its neighbor are in a defined state. As a consequence the matrix must be explored more times, from left to right and from right to left, each column from up to down and then from down to up. This is not the most logical approach, which is to switch every magnets starting from inputs to outputs. This approach does not requires multiply matrix explorations but it requires the use of many "if...then...else" and so it is much slower.

### 6.7.7 Magnetization calculation algorithm

During the matrix exploration for each cell, if a magnet is present its magnetization is calculated. Since this simulator engine is not based on a physical equation but on a tristate approximation, the calculation of the state of a magnet is done considering only the 8 cells that surround the magnet (Figure 6.24). The contribution of magnets placed at higher distance is so small that can be neglected in case of a behavioral model. The magnetization is calculated assigning at each possible state of

the magnets ("Up", "Down", "RESET") a numeric value. The total magnetization is the weighted sum of the magnetization of the neighbor elements.

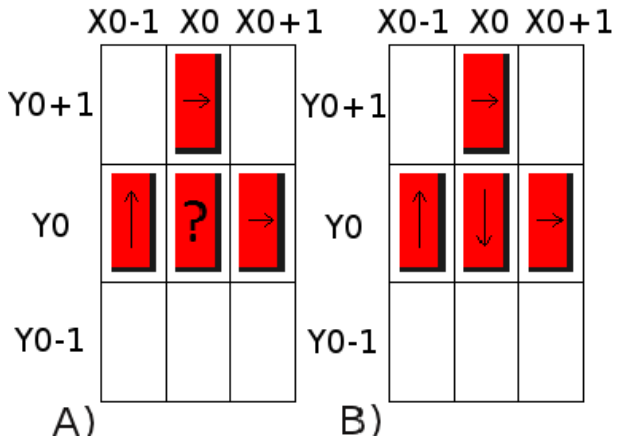



Figure 6.24. Magnet state calculation. Only the 8 neighbor cells are considered. *M.Vacca et al."ToPoliNano: A synthesis and simulation tool for NML circuits", International Conference on Nanotechnology, 2012*

To better understand how the simulation engine works in Figure 6.25 a step by step simulation of a clock zone composed only by wires is shown. A) Borders are calculated (Figure 6.25.A) and in this case they correspond to the west border because it is a case of forward propagation. B) The first column is calculated (Figure 6.25.B). This is a very simple case because for each magnet there is only one neighbor in the stable state, which is also horizontally aligned. C) The calculation of the third column always starts from the top (Figure 6.25.C). The first two magnets are not evaluated because there are no neighbors horizontally or vertically aligned in the stable states. The third magnet (X=2, Y=2) is correctly calculated. D-E) The calculation continues in the down direction and the values of the others two magnets is calculated (Figures 6.25.D and 6.25.E). F) The calculation starts again from the bottom. First three magnets produce the same results obtained before. Then the value of magnets in X=2, Y=1, that previously the algorithm was unable to evaluate, is calculated (Figure 6.25.F). The reason is that now there is a neighbor magnets in one stable state. G) Finally also the last element of the column is evaluated (Figure 6.25.G). H) Figure 6.25.H shows the final results of the simulation which is the expected one.

### 6.7.8 Exception handling

This algorithm requires a small change in case of the majority voter. Figure 6.26 from A to D shows the simulation results of the majority voter if the algorithm is

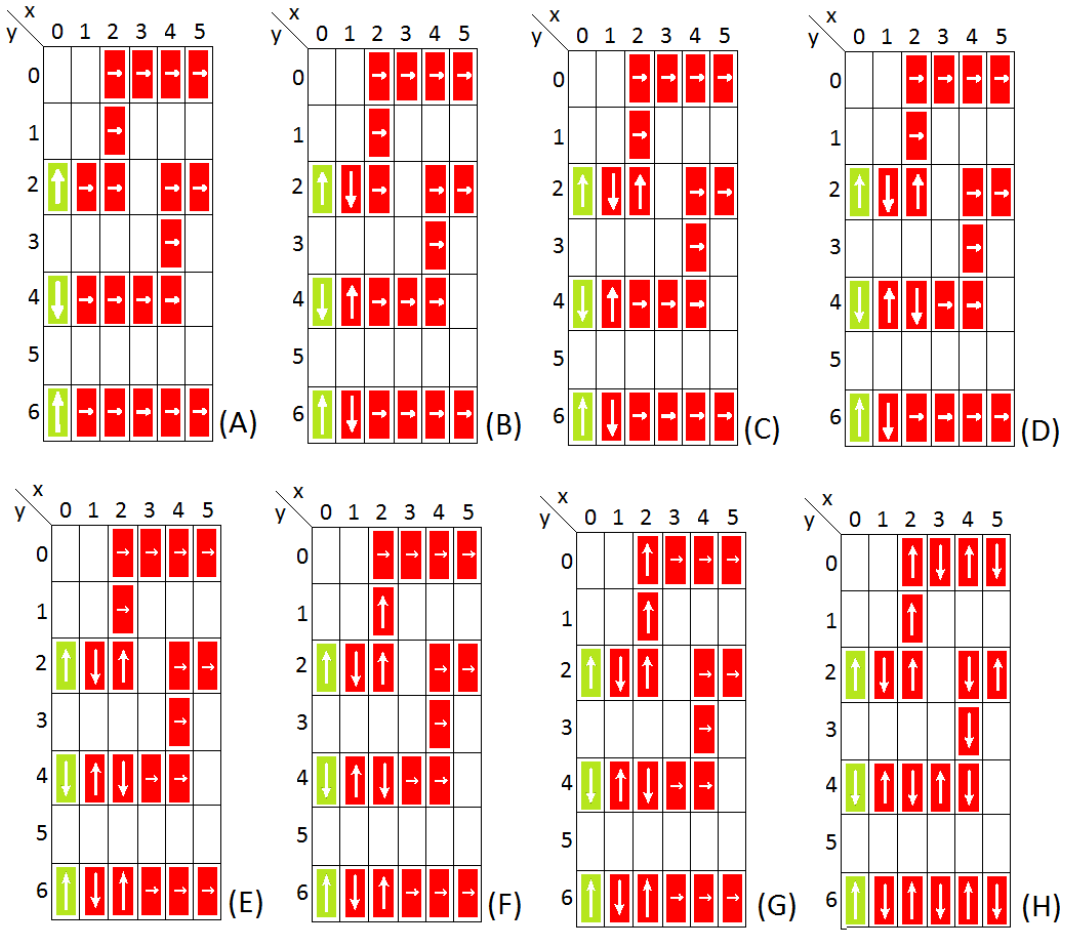



Figure 6.25. Step by step simulation of an array of three wires. *M.Vacca et al."ToPoliNano: A synthesis and simulation tool for NML circuits", International Conference on Nanotechnology, 2012*

left untouched. Magnets in (2,0) and (2,1) are calculated exactly. Magnets (2,2), which is the central magnet of the majority voter is calculated following the magnets in (1,2) because it is horizontally aligned and as a stronger influence then magnet in (2,1). Magnet in (2,3) is calculated following the central magnet of the majority voter and the last magnet of the column (2,4) is evaluated following the horizontally coupled magnet in (1,4) (Figure 6.26.B). Then the calculation starts again from the bottom. However, the re-evaluation of magnets in (2,3) gives as a result undefined (Figure 6.26.C). This happens because the magnet has two neighbors vertically coupled in (2,4) and (2,2), pointing in opposite directions. The same happens for

magnet in (2,1) (Figure 6.26.D). The problem is solved not considering the magnetization of the central element of the majority voter in the evaluation of the neighbors magnets (in positions (2,1), (1,2) and (2,3)). As a consequence, first three magnets are evaluated similarly to what happened before (Figure 6.26.E). Then, during the calculation of magnet in (2,3) the central magnet (2,2) is not considered. As a result the value of magnet (2,3) remains undefined (Figure 6.26.F). The algorithm proceeds to the bottom and restart from there. When magnets in (2,3) is evaluated again it can be evaluated correctly because magnet in (2,4) was calculated before and the central magnet of the majority voter (2,2) has no influence (Figure 6.26.G). Finally the central magnet (2,2) is re-evaluated and now the result of the whole simulation is correct (Figure 6.26.H).

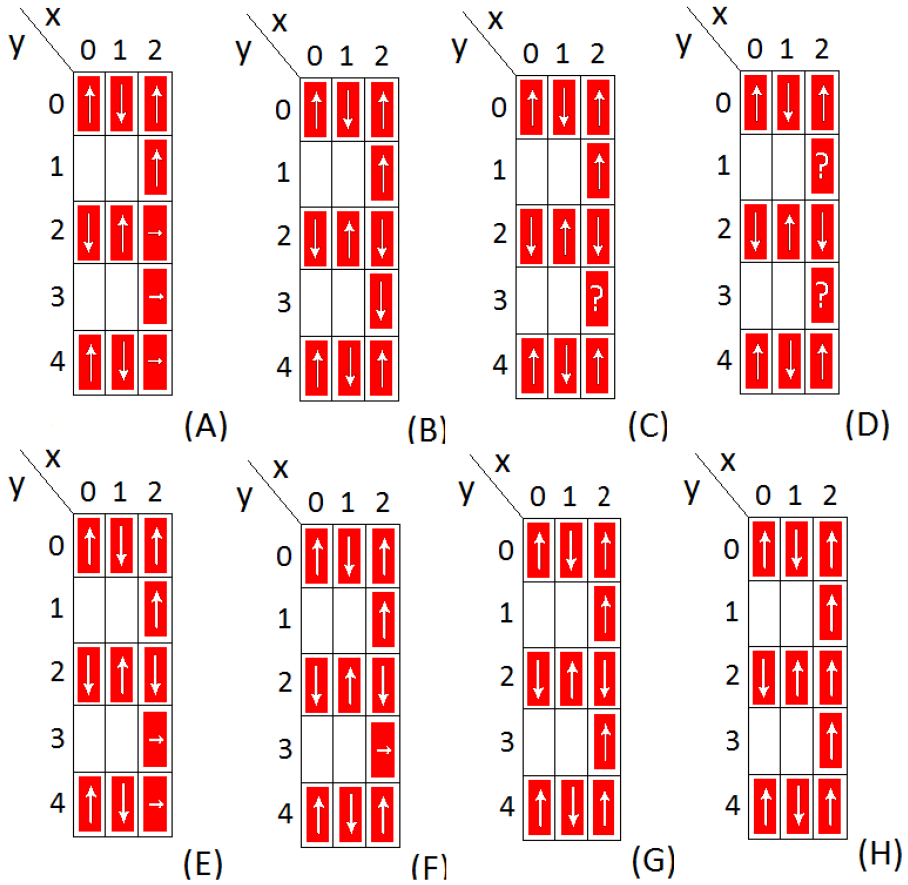



Figure 6.26. Step by step simulation of the majority voter.

Crosswires and inverters requires a slightly change to the engine. Crosswire is represented (the image that appear in the circuit layout) by 45 degrees aligned magnets, but inside the tool it is seen as a box composed by four magnets, one at each

corner. The state of the crosswire is simply evaluated assigning the same magnetization value to the magnets placed on the same diagonal. A similar representation is used for inverters. Its layout appears like a wire made by an odd number of elements, but the internal representation of the tool is a box with two magnets inside, one at the beginning and one at the end. The simulation is handled similarly to the crosswire, the value of the output magnet is forced equal to the value of the input magnet.

Also AND/OR gates require a small change in the simulation engine. The central magnet of each gate has a default value of magnetization assigned. This means that in the SWITCH phase, they will go in a defined state also if no neighbor magnets are present. Thanks to this default magnetization value, when the state of the central magnet is calculated doing the weighted sum of the magnetization of all its neighbors and itself default magnetization, only a proper inputs configuration will cause the AND gate to switch in a state different by its default value. For an AND gate its state will be "Up" only if both inputs are in the "Up" state. For OR gate its state will be "Down" only if both inputs are in the "Down" state.

### 6.7.9 Output generation

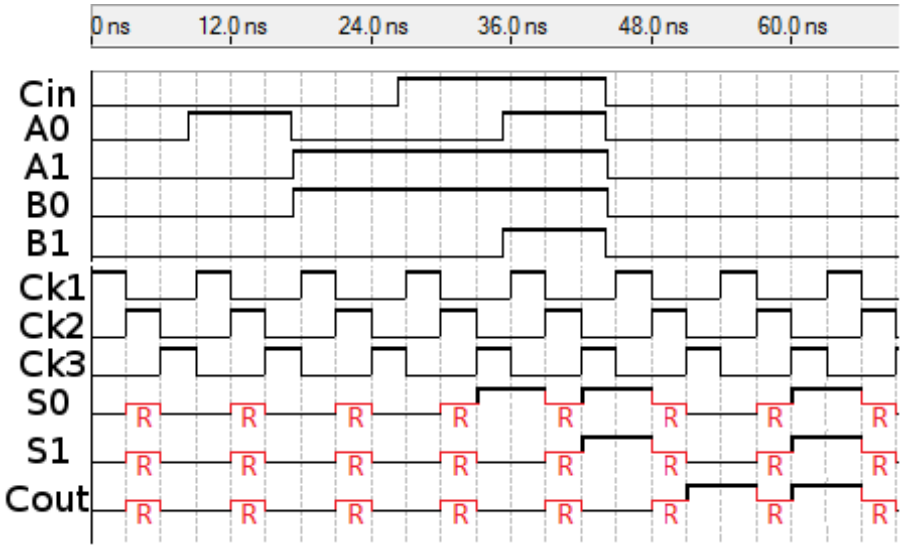



Figure 6.27. Example of a simulation waveforms of a 2 bit ripple carry adder, obtained using the full adders shown in Figure 6.6 right detail. *M.Vacca et al."ToPoliNano: A synthesis and simulation tool for NML circuits", International Conference on Nanotechnology, 2012*

Figure 6.27 shows an example of simulation obtained from ToPoliNano, considering a 2 bit ripple carry adder made using the full adders shown in Figure 6.6 right detail. As can be noted the circuit operations are correct. The overall latency of the circuit is near 3 clock cycles, since the input is in clock zone 1, the output is in clock zone 2 and the number of clock zones between them is 8. It is important to underline that no graphic waveforms generator is present up to now. ToPoliNano generates a time series of numbers written on a text file which represent the value of the outputs at each simulation step. These data must be postprocessed to obtain a graphical representation of the waveform.

# Part II

# Technological analysis

# Chapter 7

# NML physic level analysis

## 7.1 Real clock signal waveform

In NML logic as well as in QCA technology a clock mechanism is required to propagate signals trough the circuits, as explained in Chapter 1. In NML logic different techniques are possible to clock magnets, but, up to now the only one demonstrated both theoretically than experimentally is based on a magnetic field generated by a current flowing under magnets plane. The magnetic field waveform is theoretically a square wave as shown in Figure 7.1. Following this clock signals magnets are forced in the RESET state after the application of a strong magnetic field, around 100000 A/m. When the magnetic field is removed magnets align themselves to reach the minimum energy state, propagating therefore the information through the circuit. It is worth underlining that this signal is an ideal step, while the real clock signal should be a ramp as shown in Figure 7.1 (dashed line). However this necessity arise for reasons that are not strictly related to the behavior of the circuit, but to technological constraints.

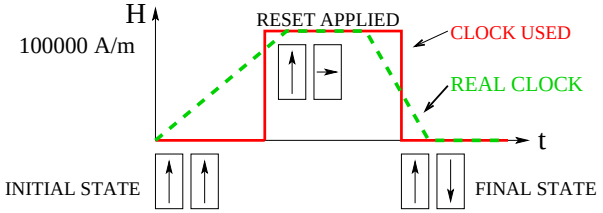



Figure 7.1. Real clock signal waveform and ideal clock signal waveforms. *M. Vacca et al. "Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

The energy required to switch nanomagnets in NML technology is extremely low, but only if the so-called adiabatic switching is used [26]. Adiabatic switching means

that the magnetic field must be applied and removed slowly, as a consequence clock signals must be ramps with a relatively long rise time (8-10 ns) [35]. Further details on this point will be given in section 7.2.1. The fall time is instead related to the number of magnets in a clock zone. As shown in [39], due to thermal noise if the number of magnets inside a clock zone is bigger than 5, a long fall time is necessary to assure that magnets switching occurs with a reduced error probability. If the number of magnets in a clock zone is equal or lower than 5 an abrupt switching, that means a clock signal with a very short (100 ps) fall time, can be used. The main consequence is that, if a circuit studied is small enough, for example a logic gate, the ideal clock waveform (step) can be used in place of the real clock signal (ramp). This is an important simplification, because detailed analysis on basic circuits must be done using low level physical simulators that are very slow. Using an ideal clock signal allows to simplify and considerably speed up the simulations.

## 7.2 Energy considerations

There are two main contributions to power consumption in NML technology: clock system losses and intrinsic energy consumption necessary to force magnets in the RESET state. Clock system losses represent the major contribution to power consumption in NML logic. A method to evaluate them in NML circuits clocked through a magnetic field is shown in Chapter 2. A solution to greatly reduce them is instead shown in Chapter 8, where an innovative clock system is proposed and compared to existing solutions and CMOS technology. As a consequence in the following, only the intrinsic energy necessary to force magnets in the RESET state is considered.

### 7.2.1 Nanomagnets switching energy

The intrinsic energy consumption depends on the energy barrier of the magnets. The energy barrier is the difference between the magnets energy in the unstable state and the magnet energy in one of the two stable states. The intrinsic energy consumption is equal to the value of the energy barrier (multiplied for the total number of magnets) if an abrupt switching is adopted. A very short rise time for the external magnetic field (one hundred picoseconds) allows the magnets to be forced into the RESET state correctly, but the energy necessary to switch the magnets is equal to the whole energy barrier (which can be in the order of $1200K_bT$. If, on the contrary, an adiabatic switching is adopted, which means a rise time of at least 8-10 ns, the intrinsic energy consumption is smaller than the energy barrier [35]. If the rise time is increased, the energy consumption decreases, and it can be reduced until it reaches the minimum value of $30K_bT$ [35], independently from the original value of the energy barrier. Energy cannot be reduced under this limit, due

to the influence of thermal noise. If the energy barrier is reduced under this limits the thermal noise will reduce the stability of magnetization vector in the logic '0' or '1' state. Magnetization will start therefore to rotate randomly also without an external magnetic field applied. This is called Superparamagnetic effect and must be avoided keeping the value of the energy barrier bigger than the $30K_bT$.

## 7.3 Errors in signal propagation due to misaligned dots

Since the RESET state is an unstable state, there can be errors in the signals propagation due to a particular alignment of nanomagnets. This kind of errors are due to the magnetic interaction among magnets when they are in the RESET state. The most important point that must be remembered is that, the circuit will always try to reach the minimum energy. This is explained in the following with the support of Figure 7.2. If two magnets are forced in the unstable RESET state, but they are perfectly aligned (Figure 7.2.A), the magnetic flux lines (schematically represented by the lines in Figure 7.2) are perfectly symmetric. As a consequence they are kept in the unstable state by the presence of the neighbor dots that have the same value of magnetization. They remain in this state until one of the neighbor magnets changes to the stable state due to the presence of an input magnet, or until the circuit state is perturbed for other reasons, like external noises or random fluctuations in the state of neighbor magnets. However, if a magnet is misaligned (Figure 7.2.B) the situation is more complex. Misalignment means that the position of a magnet is shifted with respect to neighbor magnets, as for example happens in Figure 7.2.B, where two magnets are diagonally coupled. As clear from the simplified representation of the magnetic flux lines in Figure 7.2.B, the magnetic flux is not symmetric and the length of the flux lines is not as short as possible. For this reason the misaligned dots turn down, because in this situation the length of the flux lines is shorter (Figure 7.2.C). Shorter flux lines means that the global energy of the system is lower, and therefore this situation is more stable. This means that when magnets are misaligned, there is a switching that is not due to the logic signal propagation, but it is due to the influence of magnets in the reset state. This problem is normally solved [3] adding shielding (also called "helper") blocks as shown in Figure 7.2.D (in grey). Helper blocks are magnets with an aspect ratio lower than 1, normally about 0.5. Since their longer side is oriented in the same direction of the magnetic field they are always magnetized along the x-axis. Helper blocks also helps neighbor magnets to switch, reducing therefore the value of magnetic field necessary to reset magnets. The presence of helper blocks keeps magnets in the RESET state until, thanks to signal propagation, one of the magnets switch in the stable state. As

a consequence the switch is correctly associated to the signal propagation and not the influence of neighbor magnets in the RESET state. This is a common situation in NML circuit, that happens when there is an angle in a NML wire, or more in general, in every vertical interconnection.

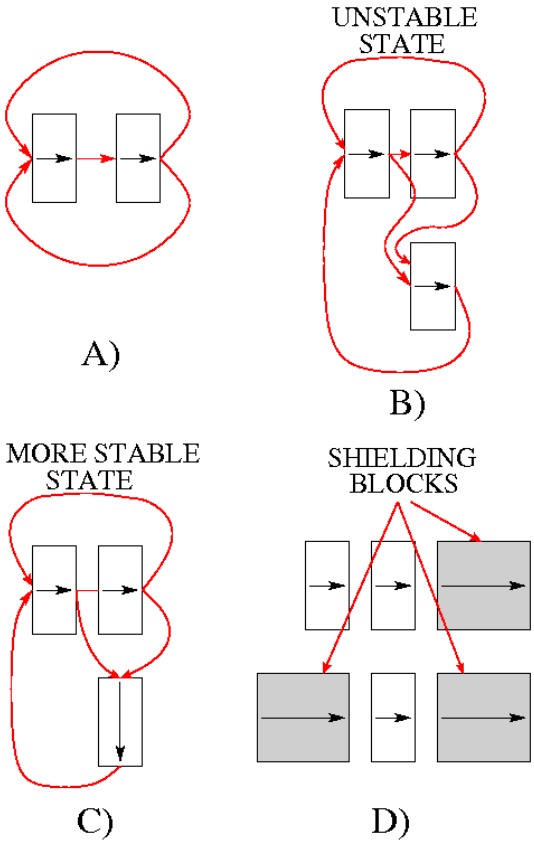



Figure 7.2. Reset problem. A) Perfectly aligned magnets. Magnets maintain the (unstable) RESET state due to the perfect alignment of the neighbors magnets. The red lines (magnetic flux) are perfectly symmetric. B) Misaligned magnets. Magnets are not in the minimum energy state. C) The misaligned element turn down due to the influence of the neighbor magnets in the RESET state. Magnetic flux lines are shorter therefore in this situation the total energy of the system is lower. D) Shielding block used to keep the misaligned elements in the RESET state, until the neighbor magnets go in a stable state. *M. Vacca et al. "Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

## 7.4 Majority voter analysis

The majority voter is the basic logic gate of this technology. It is a simple gates that allows a more complex operation than AND/OR gates normally used in CMOS. Its main characteristic is that it is a three input gate, where the value of the central magnet is equal to the value of the majority of inputs. In the follow the gate behavior is analyzed, keeping into account constraints related to the fabrication processes. The analysis is performed through low level simulations obtained using a finite element simulator called NMAG [46].

### 7.4.1 Majority voter characterization

The first task of this analysis is the verification of the correct alignment of the MV magnets magnetization when they move from the RESET to the HOLD phase through the SWITCH phase (see Figure 1.11.A), which is the first target of the simulations. Achieving the correct magnetization is not straightforward for a gate like MV, and depends on magnets shape, distances and material properties. Clearly, an incorrect alignment corresponds to a logic error. The sequence of steps are reported in figure 7.1. After the application of a strong enough horizontal magnetic field (100000 A/m), magnets are forced to assume the RESET state. The field is then removed, and magnets reach the equilibrium magnetization. For the reasons described in Section 7.1 an ideal clock signal was used in the simulations, in order to keep simulations simpler and faster. The majority voter structure used in the simulation is shown in Figure 7.3.

The majority voter structure is shown in the rectangular box in Figure 7.3. The gate has three inputs, placed on (TOP, LEFT and BOTTOM, according to figure 1.7.D) in Chapter 1. The central magnet performs the logic operation, while the block on the RIGHT is the output of the circuit. The structure of the gate is perfectly symmetric, because input signals must arrive in the same time to the central magnet, otherwise the result of the operation will not be correct. Three fixed magnetization magnets (outside the box in Figure 7.3) are used to supply the input values to the majority voter. They emulate what happens in a multiphase clock system, so they represent the last magnets of the previous clock zone. It is important to notice that when magnets are horizontally coupled there is an inversion in the signal, while vertically there is no signal inversion. Therefore the fixed magnet used to force the central input (fixed input on the left) must be inverted respect to the input that is necessary to force. If the input combination is "110", the values of the fixed (external) magnetization elements must be "100". For the same reason the value of the CENTRAL block is equal to complemented value of the majority among the three inputs. The value of the OUTPUT block (on the right) is instead equal to the value of the majority of the inputs. In Figure 7.3 the relaxed state of the

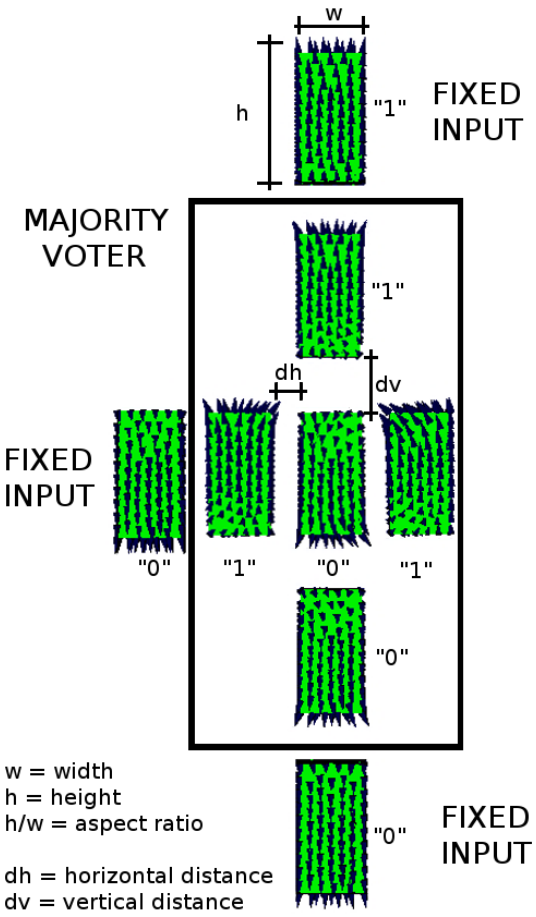



Figure 7.3. Majority Voter configuration. Fixed magnets are used as inputs for the Majority Voter. Horizontal and vertical distances and aspect ratio are changed to verify the majority voter operating area. *M.Vacca et al."Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

structure is displayed when the inputs (internal) configuration is "110" (i.e. top input has up magnetization, central input has up magnetization, and bottom input has down magnetization). Magnets used in the simulation are 20 nm thick permalloy parallelepipeds, their width is 50 nm and their height is 100 nm.

According to literature, to obtain a properly working gate, distances among magnets must be kept as small as possible [56]. From the technological point of view this requires the use of Electron Beam Lithography which is a good research tool, but too slow to be used for mass production of chips. If NML logic chips must be fabricated for commercial purposes, optical lithography must be necessary used. Considering the value of distances required Ultra Deep Ultra Violet Lithography

is required. This lithography as a resolution limit of 32 nm. While this value is extremely small, it is bigger than the value of distance used in literature for NML circuits, which is around 15-20 nm. As the main aim of this analysis was therefore the exploration of how NML can tolerate the effect of using high end optical lithography, the gate was simulated parametrically increasing horizontal and vertical distances among neighbor magnets. Therefore, for each of the eight possible inputs configurations the gate was simulated with different values of horizontal (dh in Figure 7.3) and vertical (dv in Figure 7.3) distances. Results are shown in Figure 7.4.A (top two rows of pictures). For each input configuration a map is reported in a different graph. The input configuration is detailed on top of each picture. Each point of the map represents a combination of distances that allows the gate to behave correctly. It is possible to observe that every input configuration has a different working area. In particular, some configurations have a smaller working area than others (for example input case 001 compared to case 111).

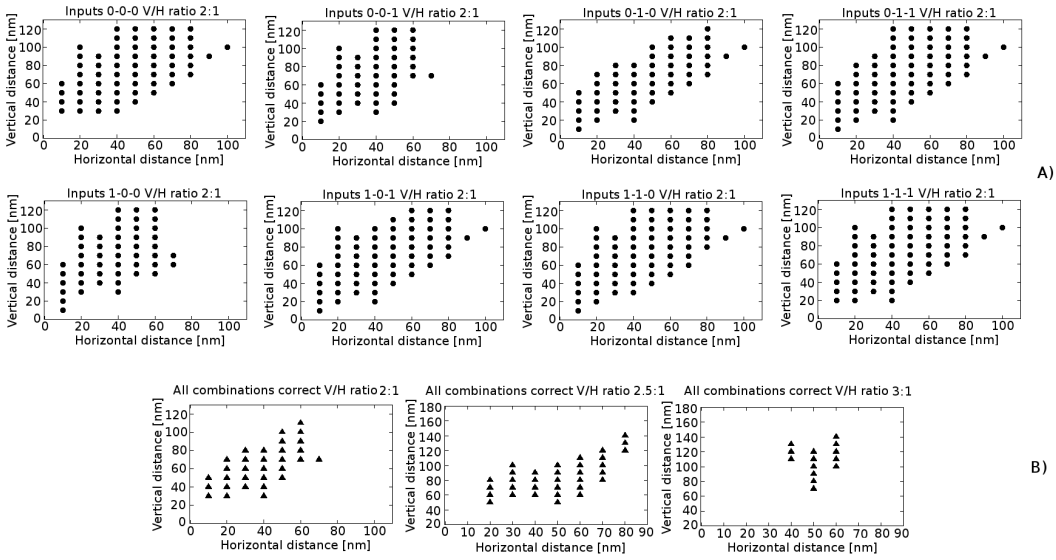



Figure 7.4. Majority voter working area with the variation of the horizontal and vertical distance. A) Working area for every inputs configuration. B) Complete working area with magnets with an aspect ratio of 2, 2.5 and 3. *M.Vacca et al."Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

The different working area of the majority voter with different inputs configurations derives from the problem exposed in Section 7.3. This is a typical example of what happens when there are two magnets are not perfectly aligned. The LEFT input magnet is diagonally coupled with the TOP and BOTTOM magnets, which are then diagonally coupled with the OUTPUT magnet. This is not a direct coupling,

but it is a case of crosstalk, an unwanted coupling between magnets. As stated is Section 7.3 to solve this problem helper blocks can be used however they tend to slow down circuit operations, therefore they were not used here as the aim of these simulations was to verify the maximum circuit speed (see section 7.4.4). The consequence of the problem described in Section 7.3 is that one of the two states ('0' or '1') is easier to reach than the other. This fact explains why there are differences in the operating area of the majority voter with different inputs configurations.

The complete working area of the MV can be obtained merging all the maps of Figure 7.4.A together. The working area is reported in Figure 7.4.B. The most important result that can be seen from this results is that the gate behave correctly also if the distance among magnets is relatively big, up to 50-60 nm of horizontal and vertical distances. This is a very important results because magnetic dots with distances of 50 nm where already obtained using Ultra Deep Ultraviolet Lithography [57]. The results here presented constitute an important breakthrough because they demonstrate that NML circuits can be fabricated with commercial-friendly fabrication techniques.

One important magnet characteristic is the aspect ratio, i.e. the ratio between the vertical height (h) and the horizontal width (w) of the magnets (Figure 7.3). Micromagnetic simulations here performed demonstrates that, at least in the MV case, it seems more convenient, in order to tolerate variations, not to reduce the aspect ration below 2. However it can be increased, with the byproduct of increasing the noise immunity of the magnets (although it is good even with an aspect ratio of 2, see section 7.4.5), and making the fabrication of dots easier since they are bigger. The same simulations performed for the aspect ratio of 2 were repeated for an aspect ratio of 2.5 and 3. In figure 7.4.B, central and right pictures report the merged working area of all inputs configurations for the aspect ratio of 2.5 and 3, respectively. For an aspect ratio of 2.5 the working area of the majority voter is similar or slightly bigger with respect to the 2 a.r., while for bigger increments the working area is greatly reduced. This happens because the magnets energy barrier depends on their aspect ratio and their volume (see section 7.4.5 for further details). Therefore if the aspect ratio is changed the magnetic interaction among neighbor magnets is drastically altered.

### 7.4.2 Impact of process variation

Since the magnetic interaction among magnets strongly depends on magnets distances and sizes, process variations may have a notable influence on the gate behavior. The process variations considered here are related to changes in magnet sizes. In particular, magnets width and height can be different with respect to the one defined at the design stage. Two types of process variations were analyzed: local mask, i.e. substrate defects that lead to differences in sizes of only one magnet of the

MV, and global errors, like under/over etching that leads to same sizes variation for all the magnets together. All possible combinations of width and height of magnets were considered in this analysis, from few tenths of nanometers to the maximum possible sizes in which magnets are merged with their neighbors. Simulations are performed considering an aspect ratio of 2 and using the 001 input configuration, which is the most critical as noticed in Figure 7.4.A. Results are shown in Figure 7.5. Each map represents a combination of widths and lengths that correspond to a proper gate operation. Figure 7.5.A shows the impact of sizes variation only of the MV LEFT input magnet. The gate still correctly operates in the 30-100 nm range for the width and in a 60-180 nm range for the height. However, results in Figure 7.5.A clearly show that the aspect ratio should better remain near 2 (the straight line on the map) or higher for a good rejection to process variations. With smaller aspect ratios the gate does works correctly only in a limited set of combinations. Figure 7.5.B, instead, shows the influence of the sizes variation of the DOWN input magnet on the whole gate behavior. Differently from the previous case, here the gate is not influenced by the aspect ratio. It works with all the width values, provided that the height is at least 100 nm. This means that in case of the DOWN magnet the key factor is the height and not the shape of the magnet, because with sizes of 120 nm width and 120 nm height the magnet is a square instead of a rectangle. The influence of the UP input magnet is not reported because it has the same behavior of the DOWN magnet, due to the symmetry of the structure. Figure 7.5.C shows the effects of sizes variation of the central magnet, which is responsible for the logic computation. In this case the MV is more sensitive to process variations. Indeed, it does not work with too high width values. Moreover, sizes of the magnet can change, but the aspect ratio should remain around 2, or be slightly smaller, to assure the correct gate operations. Figure 7.5.D shows the influence of the same process variation applied to all the magnets together. This is a quite common case in the technological processes, as it happens for example in case of under/over etching. Variations of this kind apply in the same way to all the elements. In this case the gate is much more sensitive to process variation than in other cases. The working area is smaller and it is related to the aspect ratio. Sizes can change but again with an aspect ratio around 2 or slightly lower. Moreover, if the width of the magnets becomes too small, e.g. under 40 nm, the gate risks not to work properly.

The results here presented are very promising showing a relatively good tolerance to process variations. If the variation is not too big, logic gates still correctly work. What is it clear from this analysis is that the aspect ratio is the key factor in in-plane NML technology. Process variations that affect all the magnets are the most troublesome, but fortunately they can be compensated quite well by correctly setting up the technological process. These results demonstrate that the majority voter can work, automatically validating the entire technology. This happens because the other logic gates, such wires and inverters, have a simpler structure and therefore

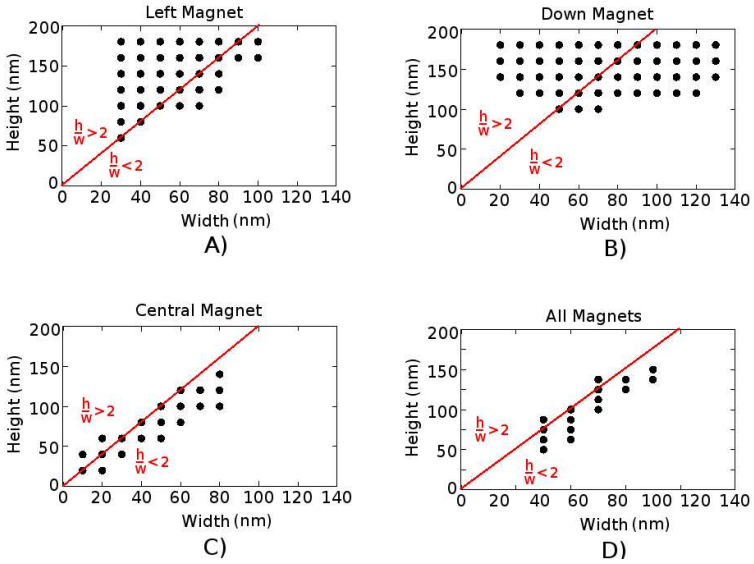



Figure 7.5. Majority voter working area considering process variations. Red line represent the aspect ratio 2. A) Sizes variation of the left magnet. B) Sizes variation of the down magnet. C) Sizes variation of the central magnet. D) Sizes variation of all the magnets together. *M. Vacca et al. "Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

they are subject to less complex magnetic interaction. So, if the MV works, also the other structures will work.

### 7.4.3 NMAG automatic C framework

A framework written C was created to easily make thousands of simulation automatically. The use of this framework is necessary because the required time, to simulate all the cases above mentioned is between 30-40 hours, depending on the machine used.

This framework is a program written in C which code is reported in the appendix C. The behavior of the framework is quite easy to understand looking at the main file (Section C.1 in Appendix C.

- First some constants are defined, which indicate the minimum and maximum sizes of the magnets and the step used for the increment.
- Then for each value of magnet sizes the geometry file is defined, calling the function "geometry" (Section C.2 in Appendix C. This function create a file which describes the geometry. Starting from the geometry file the mesh is created using the program "NETGEN", called from the command line.

- Now the main simulation file is executed "nsim mv.py" (Section C.3 in Appendix C.
- After this simulation results are converted in a more comfortable file format using the built in script "ncol" of NMAG.
- The results are now plotted to graphs executing "nsim graph.py" (Section C.4 in Appendix C.
- Finally unused files are deleted and the obtained results are moved to an appositely created folder.
- This is automatically repeated for each possible value of magnet sizes.

### 7.4.4 Timing analysis

As explained before the timing evolution of the circuit is not strictly related to the magnetodynamics, but is mainly related to other constraints, like the necessity to have a low power consumption using an adiabatic switching or to avoid errors in the signal propagation. However it is important to evaluate the impact of the change in magnets distances, to better understand the dynamics of NML circuits. To evaluate the timing performance of the MV, during the previous simulation the 50% delay was measured. In case of NML technology the 50% delay is the delay between the 50% of the variation of the clock signal (during the fall ramp when the switching can start) and the 50% of the variation of the magnetization of the CENTRAL block. The 50% delay was chosen because it is similar to the reference time used in CMOS circuits, so an easy comparison between the two technologies can be obtain. Figure 7.6 shows an example of MV dynamics obtained postprocessing NMAG results. Figure 7.6 shows the variation of the CENTRAL block with the time.

For all values of horizontal and vertical distance that follow the maps of Figure 7.4 the 50% delay was measured, only considering the aspect ratio of 2. Figure 7.6 shows an example of waveforms obtained postprocessing NMAG results. In this example the horizontal distance is fixed at 20 nm (the first number after the 'M' near each curve), and for different values of vertical distance (the second number near each curve), considering the input configuration 010. The 50% delay of the gate is of the order of hundreds of picoseconds and it increases with the increase of distances among magnets. The reason behind this fact is easy to understand, increasing the distance among magnets reduces the magnetic coupling between neighbor magnets. If distances are too big the magnetic field generated by a magnet is not strong enough to force its neighbor in the correct state. In this case for example, with a distance of 80 nm, the results is wrong, according to the map of Figure 7.4.A.

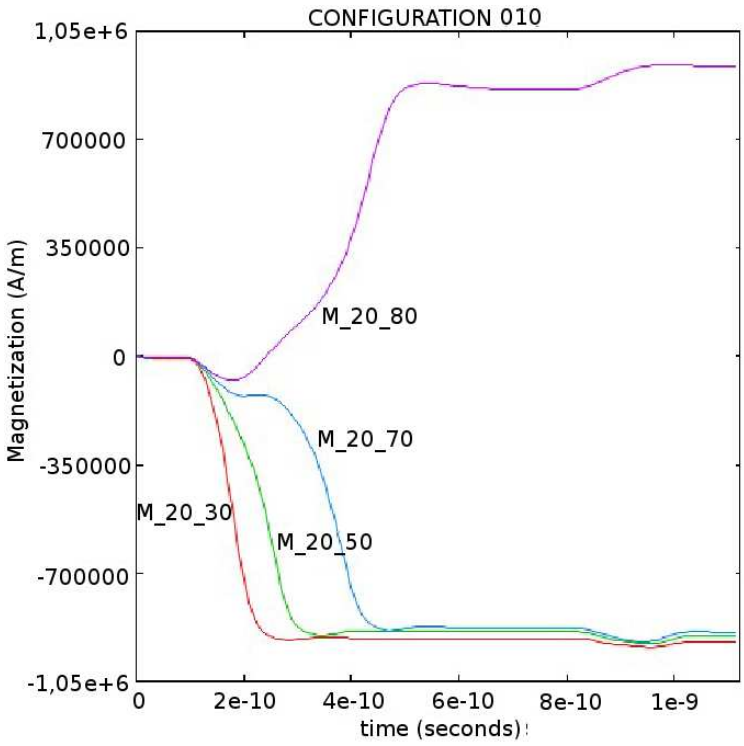



Figure 7.6. Timing variation of the central magnet magnetization in a few cases of vertical and horizontal distance for the input configuration of 010. The different waveforms identify different values of horizontal and vertical distance. The first number represents the horizontal distance while the second number identifies the vertical distance. Different waveforms are presented: In the first three the gate works properly, and in the last one the behavior of the gate is wrong as magnetization is expected to go to a negative value (which represents logic 0) but goes to a positive value (which represents logic 1). *M. Vacca et al. "Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

Table 7.7 expands the results of Figure 7.6 showing the 50% delay with the same values of distances, 20 nm horizontally and 30 nm, 50 nm, 70 nm vertically, for each of the eight inputs configurations. The variation of the delay with the increment of vertical distance depends on the input configuration. With some input configurations the delay shows small variations (considering the relatively high tolerance on the measured values), while with other input configurations the delay considerably increases. This behavior is due to the "reset problem" explained in section 7.3.

To better figure out the relations between MV timing and distance variations, values can be properly grouped together. Figure 7.8 shows, for each value of horizontal distance (x axis), the minimum and maximum delays measured among the

| Inputs | Vertical Distances (nm) | | |
|---|---|---|---|
| ABC | 30 | 50 | 70 |
| 000 | 90ps | 130ps | 160ps |
| 001 | 230ps | 130ps | 210ps |
| 010 | 100ps | 140ps | 260ps |
| 011 | 210ps | 200ps | 200ps |
| 100 | 240ps | 230ps | 210ps |
| 101 | 100ps | 120ps | 220ps |
| 110 | 210ps | 200ps | 200ps |
| 111 | 110ps | 120ps | 150ps |

Figure 7.7. Timing variation with three values of vertical distance for the each input configuration, considering an horizontal distance of 20 nm. *M.Vacca et al."Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

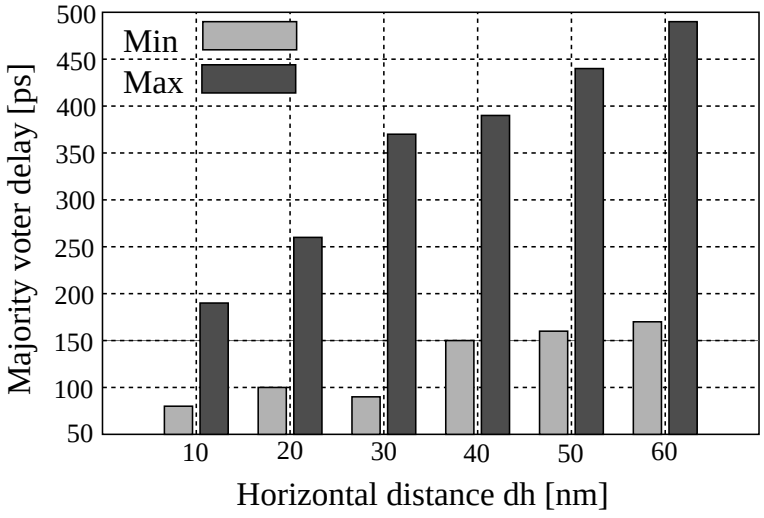



Figure 7.8. Timing variation of the gate. For each value of horizontal distance the minimum and maximum values of delay, measured among all the input configurations and all the vertical distance, are reported. *M.Vacca et al."Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

simulations obtained changing all the possible input configurations and all the values of vertical distances. Results show that the average value is between 100 ps and 300 ps, and it increases with the increment of horizontal distance. As a consequence

of this analysis we can conclude that distances must be kept as small as possible, compatible with the technology, in order to improve the overall circuit speed. However, it is worth underline again that the clock frequency does not depend on the gate delay, because it is determined by others factors: The long rise time necessary for adiabatic switching to reduce power consumption, the high fall time in case of more then 5 magnets for clock zones and also the necessity to use a three phase overlapped clock system. The delay determined here represents the lower bound of this technology, and it will lead to a maximum allowed clock frequency of about 1 GHz. Considering all the other constraints, the obtainable clock frequency is expected to be between 10 MHz and 100 MHz [25].

### 7.4.5 Energy analysis

To understand the dynamics of a NML circuit it is important to understand how energy is related to magnets sizes and distances. It is true that the power consumption of the energy depends also on other factors, like the use of an adiabatic switching, however to obtain a complete picture of the technology it is necessary to evaluate the energy barrier of magnets. NMAG allows the evaluation of the energy components related to magnets dynamics. So for each value of horizontal and vertical distance, following the maps of Figure 7.4, it is possible to evaluate the average energy barrier of the majority voter. Figure 7.9 shows the results of the simulations. The minimum and maximum value of energy are reported for each horizontal distance value (x-axis). The minimum value of barrier is obtained when the vertical distance is maximized, while the maximum energy value is obtained for the minimum value of vertical distance. Energy increases with the horizontal distance however it saturates at a value of around 50 nm.

This behavior can be easily explained considering how the energy barrier is composed. There are two main components that compose the energy barrier of a magnet. The first component is the Demagnetization energy, which depends on the magnets volume, its aspect ratio and the type of magnetic material chosen. The second component is the Exchange energy which is a quantum term that describes the interaction between two neighbor magnets. It is a quantum quantity so it only counts when magnets are very close. The Exchange energy is also related to the reciprocal magnetization of neighbor dots, so it can increase or decrease the magnets energy barrier. Particularly when two magnets are antiferromagnetically coupled (with magnetization pointing in opposite directions, like in the horizontal coupling) the Exchange energy reduces the value of the energy barrier, so when the horizontal distance is reduced the effect of the Exchange energy decreases the value of energy barrier. When magnet are coupled ferromagnetically (with magnetization pointing in the same direction, like in the vertical coupling) the Exchange energy increases the value of the energy barrier, therefore reducing the distance among magnets increases

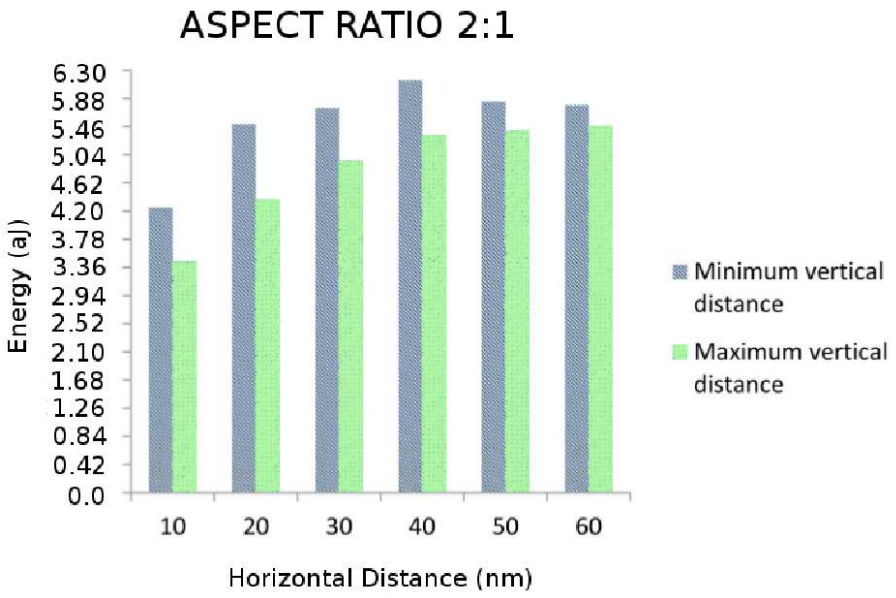



Figure 7.9. Power analysis with all the possible inputs configurations, for all the vertical and horizontal distance values with an aspect ratio of 2. *M.Vacca et al."Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

the energy barrier. The value of energy barrier can be reduced if the horizontal distance decreased and the vertical distance increased. When distances increase too much the contribution of the exchange energy drops to zero, and the value of the energy barrier becomes constant and equal to the demagnetization energy.

Figure 7.10 shows the variation of the energy barrier considering an aspect ratio of 2.5. The general trend is the same but the absolute values are different with respect to the values found for the aspect ratio of 2. With an horizontal distance of 50 nm the energy barrier increases from 5.2 aJ to 8 aJ. This happens because the value of the demagnetization energy depends on the volume of the magnets but also on the aspect ratio. Increasing the aspect ratio the energy barrier rises, and this means higher noise immunity in the stable state. However this also causes higher power consumption if an abrupt switching is adopted, or means lower clock frequency in case adiabatic switching is the choice.

Results are similar if the aspect ratio of 3 is used (Figure 7.11). The general trend is again the same but the absolute value is notably risen. With a width of 50 nm the value of the energy barrier is 11 aJ. This means that with an increment of the aspect ratio from 2 to 3 (50%) the value of the energy barrier is doubled.

This energy analysis allows to extrapolate some interesting conclusions. First, the value of the energy barrier can be considered independent of magnets distances,

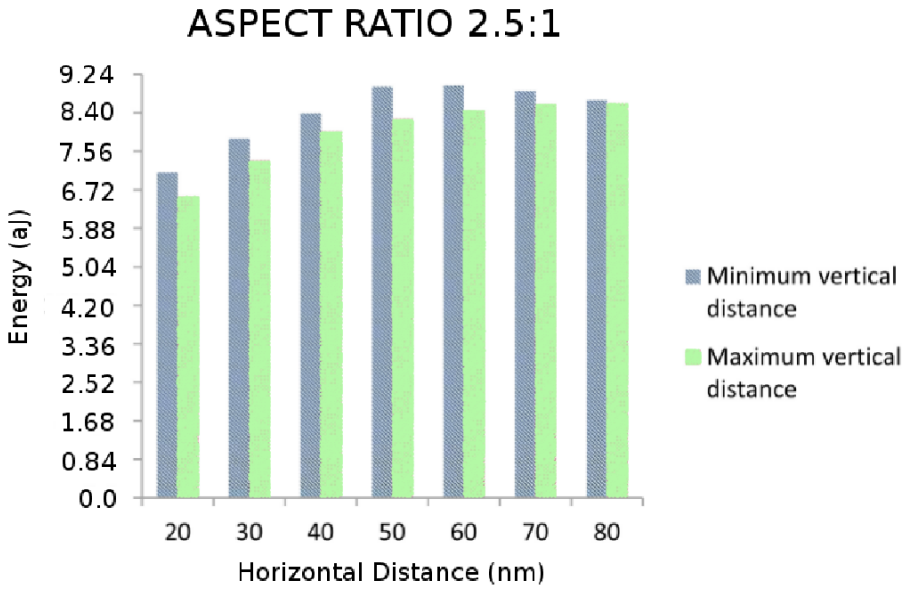



Figure 7.10. Power analysis with all the possible inputs configurations, for all the vertical and horizontal distance values with an aspect ratio of 2.5. *M. Vacca et al. "Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

if magnets are fabricated using Deep UV lithography, which means distances of 40-50 nm. Second, two approaches toward NML are possible. An adiabatic switching makes the switching energy independent from the value of energy barrier, however the rise time depends on the value of energy barrier, so the smaller the energy barrier is, the faster the circuit is. If an abrupt switching is used the circuit speed increase, however the switching energy is equal to the entire energy barrier, so reducing the value of the energy barrier reduces the energy consumption. In both cases the conclusion is that it is important to keep the aspect ratio as small as possible. By increasing the aspect ratio, the noise immunity also is improved, but with an aspect ratio of 2 the noise immunity is quite high as well. An energy barrier of 5,2 aJ, for example, corresponds to 1250 KbT, which is much higher than the value of 40 KbT, the minimum value necessary to assure the thermal stability and a low error probability. The best solution is therefore an aspect ratio of 2 or slightly smaller. A final note can be done on magnet sizes. If technology allows it, in case magnet sizes are reduced at least to a width of 15 nm, a height of 30 nm height, and a thickness of 5 nm, the value of the energy barrier decreases to 40 KbT. In this case it is possible to use an abrupt switching, obtaining clocking frequency of 1 GHz, and thus minimizing at the same time power consumption.

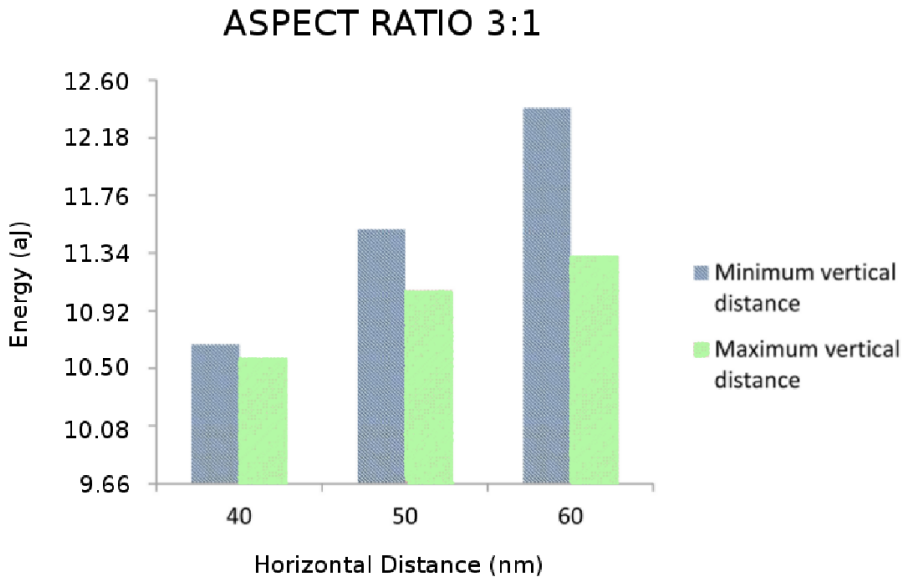



Figure 7.11. Power analysis with all the possible inputs configurations, for all the vertical and horizontal distance values with an aspect ratio of 3. *M. Vacca et al. "Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

### 7.4.6 Majority voter input extension

#### Majority voter not working with linear clock wires

The classic MV analyzed here requires that all the inputs arrive at the same time. For this to happen the clock zone should be limited to exactly the size of the majority voter. However a feasible normal clock signal is generated using parallel wires placed under the plane of the magnets. In this case, inputs are required to come from the same direction as shown in Figure 7.12.A. The top picture shows the structure in a simple sketch, the bottom picture shows the result obtained by a OOMMF simulation [45] in the same configuration. This magnets organization is problematic because, while in the classic case (Figure 1.11.B) the length of every input arm of the gate is equal, in this case the length of the upper and lower arms is bigger, due to the presence of an angle. The consequence is that the left input signal arrives before the others two, and the gate does not work properly in all the configurations. Moreover, while the classic majority voter can work also without the use of shielding blocks, in this case they are mandatory. A possible solution is presented in Figure 7.12.B [45], where the length of the arms are equalized reducing the number of magnets in the upper and lower arms, placing them at an higher distance. Another alternative solution is sketched in Figure 7.12.C [45] where the number of magnets is increased

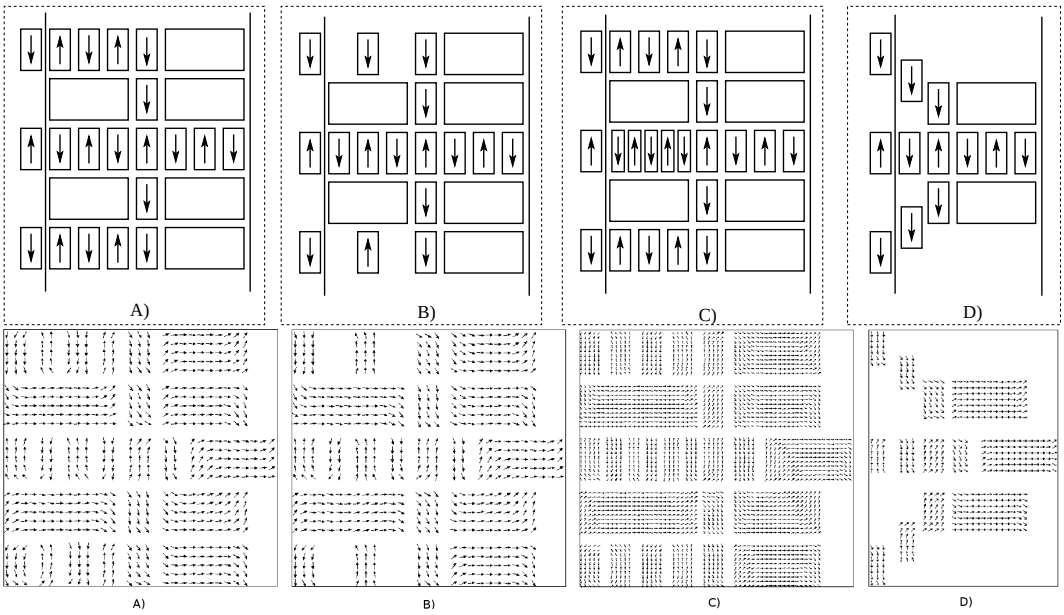



Figure 7.12. Majority voter possible solutions with inputs coming from one direction. Top line pictures: a sketch to clearly show the magnets organization and magnetization. Bottom line pictures: OOMMF simulation of the same configuration. A) Classical structure with inputs extended. B) Reduction of the number of elements in the up and down arms. C) Increment of the number of elements in the central arm, making them smaller. D) Displacement of the corner elements to equalize the number of magnets in each arm. *M.Vacca et al."Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

in the left arm, using smaller magnets. Again, simulations show that both these solutions do not give the expected results in all the configurations. The structure of NML circuits must be symmetric with magnets of the same sizes, and possibly with the same distances. A further possible solution is presented in Figure 7.12.D [45], where dots are misaligned. Simulation shows that this solution does not work due to the "reset problem" described in section 7.3, and to the impossibility to to place shielding blocks due to lack of proper space.

### Modification of clock wires shape

These simulations highlight a characteristic of NML circuits: circuits must be as much symmetric as possible, with magnets of the same sizes and with the same distances. Therefore the use of the classic majority voter with this clock system seems impossible. A possible solution is to use a AND/OR gate as presented in

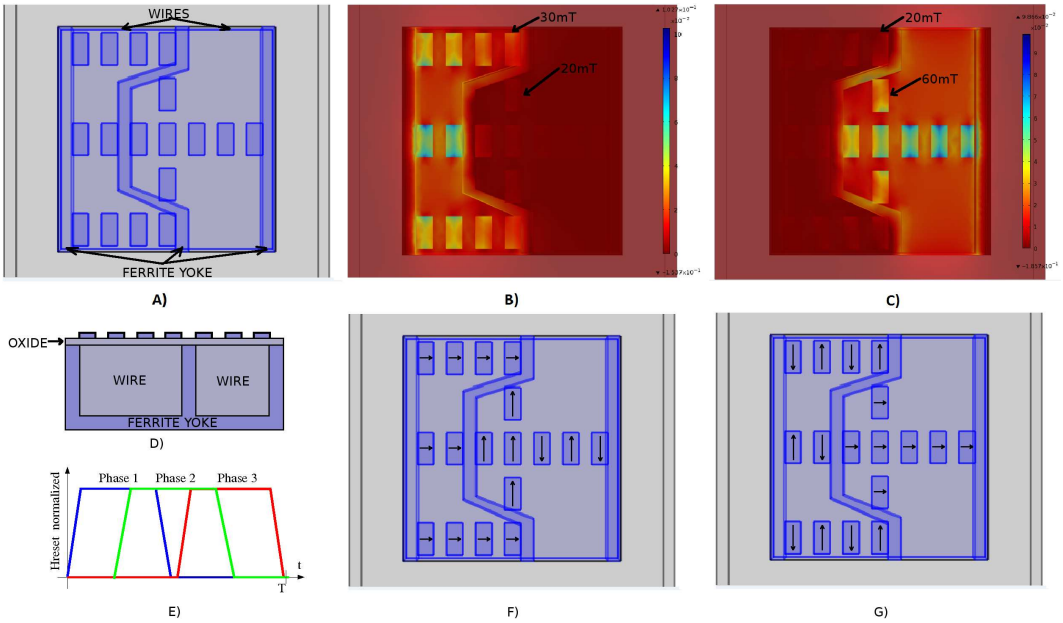



Figure 7.13. Comsol Simulation of clock wires A) Clock wires model. B) Simulation results with current flowing in the first clock wire. Color gradations represent the horizontal component of the magnetic flux density (B) expressed in Tesla. C) Simulation results with current flowing in the second clock wire. *M.Vacca et al."Majority Voter Full Characterization for Nanomagnet Logic Circuits", IEEE Transaction on Nanotechnology, 2012*

[6]. However this is an important limitation, because the use of the MV can greatly improve the set of logic gates available in the NML technology, allowing the design of more dense circuits. For this reason, here, is proposed a different solution: a local modification of the clock wires that makes possible the fabrication and the proper operations of a MV. The structure is shown in figure 7.13.A. Clock wires are shaped (routed in a different plane clearly, see [5]) around the majority voter. In this way signals arrive at the gate inputs simultaneously. The darker lines which surround the wires represent a ferrite yoke used to confine magnetic flux lines and to reduce the current necessary for magnets switching as proposed in [22] (a section view is in figure 7.13.D).
Although this solution is not of easy implementation from the technological point of view, simulations obtained using Comsol Multiphysics [58] show that the structure assures a proper MV behavior. In Figure 7.13.B the magnetic flux density is shown (top view) when the current flows through the left wire. The current values applied are such that the intensity of the magnetic field is kept at the minimum value necessary for magnets to switch. This is done to reduce clock power consumption.

Magnets of the left clock zone should be forced in the RESET state, while magnets on the right clock zone are supposed to stay in the HOLD state (see figure 7.13.F). Considering the worst case measured in the simulation, the magnetic flux density is double on the magnets of the left clock zone with respect to magnets of the right clock zone.

However, Figure 7.13.B shows that on magnets of the left clock zone placed in the corner the magnetic flux density is low, and might be too low to assure the magnet reset. To assure magnets reset an higher current should be used. However, in this case also the (peripheral) magnets of the right clock zone might reset. But, as described in Chapter 1,a three overlapped phases clock is immune to border crosstalk. This modification of clock wires, although not easy to implement, assures the possibility to fabricate properly working MVs, increasing then the set of gates available and the density for NML technology.

## 7.5 Inverter

The majority voter is the most important logic gate inside NML logic, but to build any kind of circuits they must be coupled with at least another logic gate, the inverter. In NML logic horizontal wires, i.e. a sequence of magnets aligned horizontally, signals propagate through antiferromagnetic coupling. This means that each magnet assumes the inverted value of its neighbors. In vertical wires instead there is no signal inversion, because magnets assume the same value of their neighbors. As a consequence apparently no inverter is required, because and odd number of magnets horizontally aligned performs the signal inversion. Unfortunately the situation is much more complex. The number of magnets that can be aligned horizontally depends on the width of the clock zones. If clock zones are chosen width enough to contain and even number of horizontally aligned magnets there will be no signal inversion. If the clock zone width is chosen with the aim to contain an odd number of magnets instead, all the signals inside that clock zone will be inverted. However the purpose of an inverter gate is to invert only a specific signal. Making clock zones width enough to contain an odd number of magnets is therefore useless, and a specific logic gates which performs the signal inversion is required.

Figure 7.14.A shows an horizontal NML wire made by an even number of magnets. As it is possible to see the last magnet assumes the opposite value of the first magnet. As a consequence if another element is placed after the last magnet it will assume exactly the same value of the first magnet of the wire, therefore there is no signal inversion. To build an inverter it is possible to exploit diagonal coupling between two magnets. When two magnets are coupled through their diagonal there is no signal inversion. Figure 7.14.B shows a possible inverter layout, which is the direct mapping of the inverter layout used in general QCA. Two important things

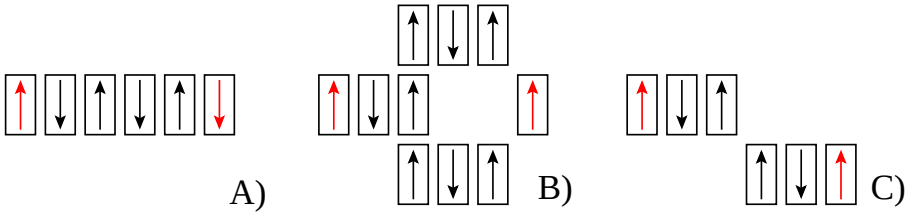



Figure 7.14. A) NML wire. The last magnet is in the opposite state of the first one, as a consequence the first magnet that is placed after the wire end will have the same value of the first magnet, as a consequence no inversion of signal is present. B) Possible inverter layout. The last magnet has the same value of the first magnet, as a consequence the next magnet in the chain will have an inverted value with respect to the first magnet. C) Simpler inverter layout.

can be noted, first diagonally aligned magnets have the same value and second the first and last magnet assume the same value. If another element is therefore placed after the last magnet of the wire, it will assume the opposite value of the wire first magnet, performing therefore signal inversion. The layout can be simplified as shown in Figure 7.14.C. The structure is simpler but there is still a signal inversion. Micromagnetic simulations of these two structures show that the gates does not work correctly. The reason lie in the problem described in Section 7.3, due to the influence of magnets in the RESET state on misaligned magnets. Other solutions must be therefore used.

The width of a clock zone must be chosen according to the number of magnets that must contain. If the width of the clock zone is 6 magnets, the maximum number of elements that can be horizontally aligned is 6. However this is only true if the spacing between neighbor magnets is assumed constant. If the space among neighbor magnets is reduced, more magnets can be placed in the same clock zone. Figure 7.15 shows an example of two wires: In the upper wire 6 magnets are aligned horizontally with a space among them of 20nm, in the lower wire 7 magnets are aligned horizontally with a space among them of 10nm. As it is possible to see the total width of the area is almost the same, so both wires can be placed in clock zones with the same width. The different is that in the upper wire there is no signal inversion, while in the lower wire there is signal inversion. The lower wire represents therefore the NML inverter. Figure 7.15 shows the micromagnetic simulation (obtained using OOMMF [45]) of the wire and the inverter, a magnet with an aspect ratio of 0.5 is used as input, while an helper block is used at the end of the circuit. In Figure 7.15.A magnets are successfully forced in the RESET state by an external magnetic field, while in Figure 7.15.B magnets realign correctly after the removal of the magnetic field. The inverter here developed is the only type of inverter that can work in NML logic. The main disadvantage is that it

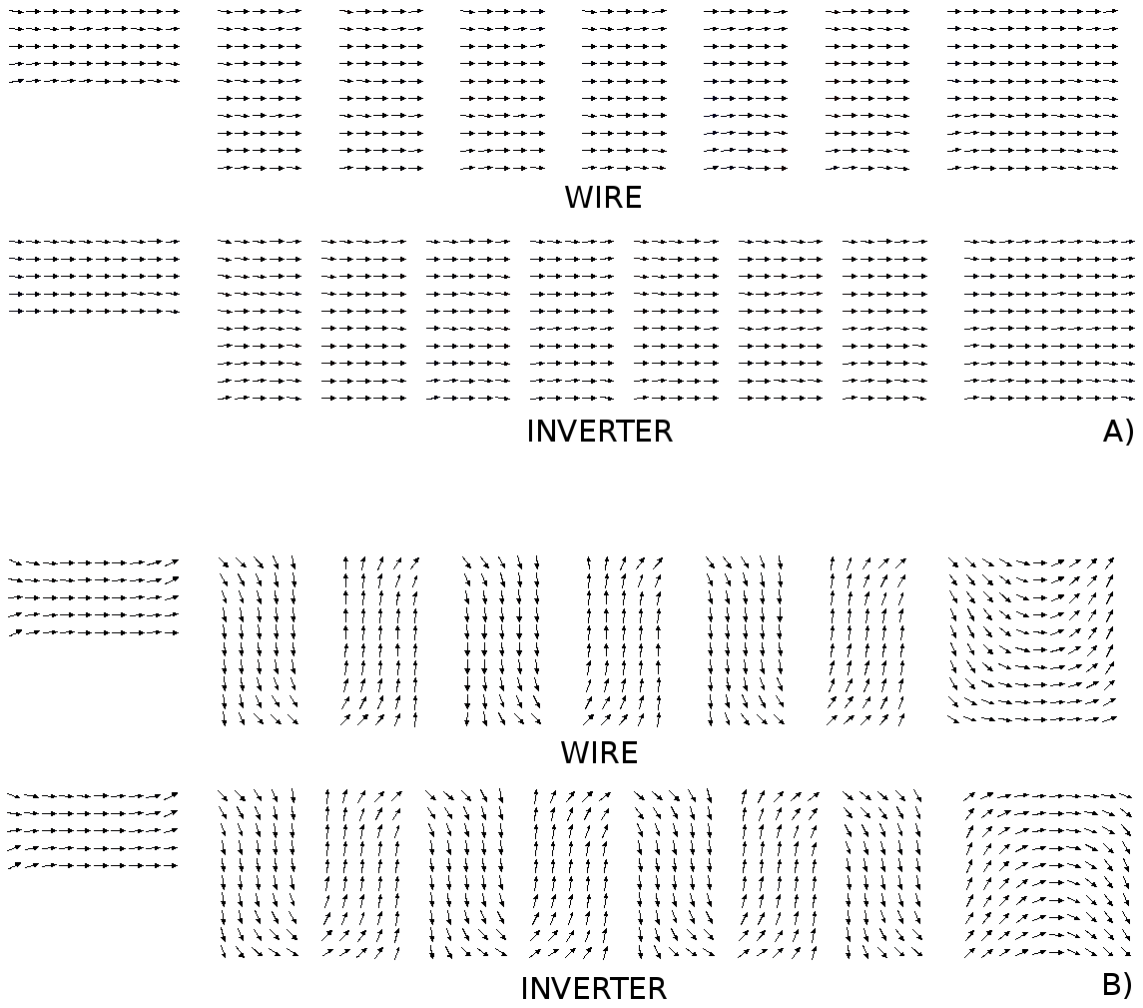



Figure 7.15. Inverter and wires low level simulation. A) Magnetic field applied. B) Magnetic field removed.

requires a lower distance among neighbor magnets, making the circuit fabrication more complex.

## 7.6 Global clock system

In [6] a two phase clock mechanism was proposed. It is described in Figure 7.16. Two clock signals are used, each of them is a square wave with duty cycle of 50% but with a phase difference of 180 degrees. Every clock zone has a magnet with a trapezoidal shape at the beginning. In the classic multiphase clock system, at least three clock phases are necessary to assure that signal propagates in a specific

direction. In the two phase clock the trapezoidal element is used to grant that there is no backward propagation, therefore two phases are enough. Thanks to their shape, trapezoidal elements have a greater influence on magnets placed near their longer side, while they have a lower influence on magnets placed near their shorter side. As a consequence when the magnetic field is removed, only magnets placed near their longer side will start to switch. In case of Figure 7.16 signals propagate therefore from left to right also using only two phases. The advantage of this clock system is that the circuit fabrication is simpler, particularly in case of feedback because no clock wire twisting is required. The disadvantage is that the latency of the circuit is increased, reducing the throughput in case loops are present.

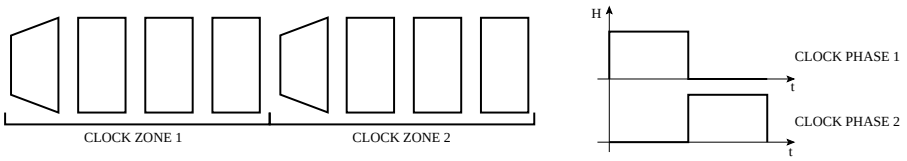



Figure 7.16. Two phase clock system. Trapezoidal magnets are used to force the signal to propagate in a specific direction.

The solution here proposed is a further exploitation of the mono-directionality due to the use of trapezoidal magnets. The idea is to totally eliminate clock phases and zones and use a global clock mechanism as happens in out-of-plane nanomagnet logic. In out-of-plane nanomagnet logic [25] cobalt-platinum square dots are used. The key mechanism in this type of NML is magnetocrystalline anisotropy, which is a different type of magnetic anisotropy related to the crystals orientation inside the material itself. Thanks to this property magnetization lies perpendicular to the plane as shown in Figure 7.17.A. in out-of-plane NML a clock signal applied globally to the circuit is used (Figure 7.17.B). No clock zones and clock phases are used, signals propagate in a specific direction thanks to how magnets are fabricated. One part of them (the gray part in Figure 7.17.A) is irradiated with an ion beam, locally changing the magnetic properties of the material. In this way magnetic coupling with dots placed near the irradiated side is weaker than coupling with dots near the not-irradiated side, and signals propagate therefore in a specific direction without the need of clock phases. This is the same mechanism of trapezoidal magnets, so the idea is to develop a similar clock system also in case of classic NML logic. Differently from Figure 7.16 circuits are built only by trapezoidal magnets. Signals propagate in a specific direction depending how magnets are coupled. In Figure 7.17.C signal propagates in the right direction, while in case of Figure 7.17.D signal propagates from right to left. The clock signal is a sinusoidal magnetic field applied in plane along the longer side of magnets. The reason behind the choice of a global clock system instead of a local clock system, are that a global clock system is much more

easy to manage. As shown in Figure 7.17.E is sufficient to generate a magnetic field applied globally to the entire chip. This can be generated for example with a on-chip solenoid. A global clock system pose less difficulties in the fabrication of the chip, and also removes all the problems related to the confinement of magnetic field in classic NML logic.

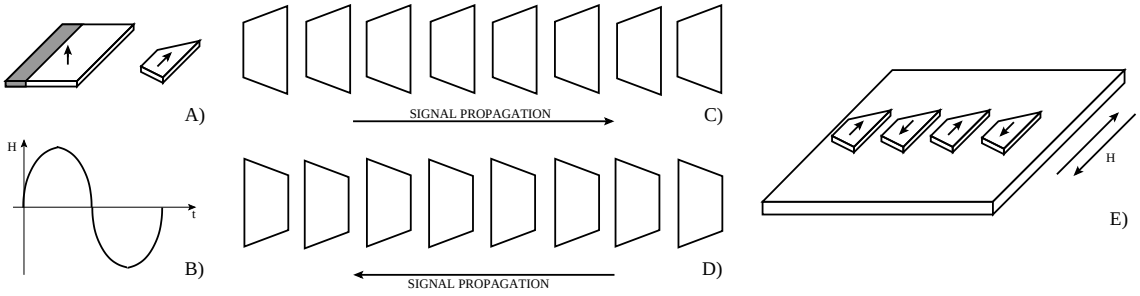



Figure 7.17. Proposed global clock system. A) In Out-of-plane NML logic magnetocrystalline anisotropy is used in place of shape anisotropy, magnetization lies therefore out-of-plane. Signal propagation direction is forced irradiating part of the dot with an ion beam (the gray part of the magnet), locally changing magnetic properties. The same thing can be obtained in classic NML changing the magnets geometry, using trapezoidal magnets. B) Global clock signal. C) Signal propagation in the right direction. D) Signal propagation in the left direction. E) Magnetic field is applied globally to the entire chip, using for example a on chip solenoid. A sinusoidal magnetic field is applied in plane along the longer side of magnets.

Figure 7.18.A shows how the global clock system should works. Its basic mechanism is much different from classic NML clock, because magnets are not forced in the intermediate unstable RESET state. The purpose of the global clock is to supply the magnets with the energy required to switch one of its neighbors, energy that alone the magnets do not posses. At the beginning (Figure 7.18.B) the input magnet is switched. When the clock signals reaches its positive maximum value, the sum of the external magnetic field and the magnetic field generated by the input magnet is strong enough to switch the second magnet from down state to up state (Figure 7.18.C). This happens because the magnetization of the input magnet points down, therefore the magnetic field generated near the second magnet points up and has the same direction of the external magnetic field. When the magnetic field reaches its maximum negative value (Figure 7.18.D), the sum of the external magnetic field (which now points down) and the magnetic field generated by the second magnet (which now points down in correspondence of the third magnet) is strong enough to switch the third magnet. When the magnetic field reaches again its maximum positive value it is the forth magnet that switches (Figure 7.18.E). Signal propagates therefore through the circuit with a domino-like effect following the time

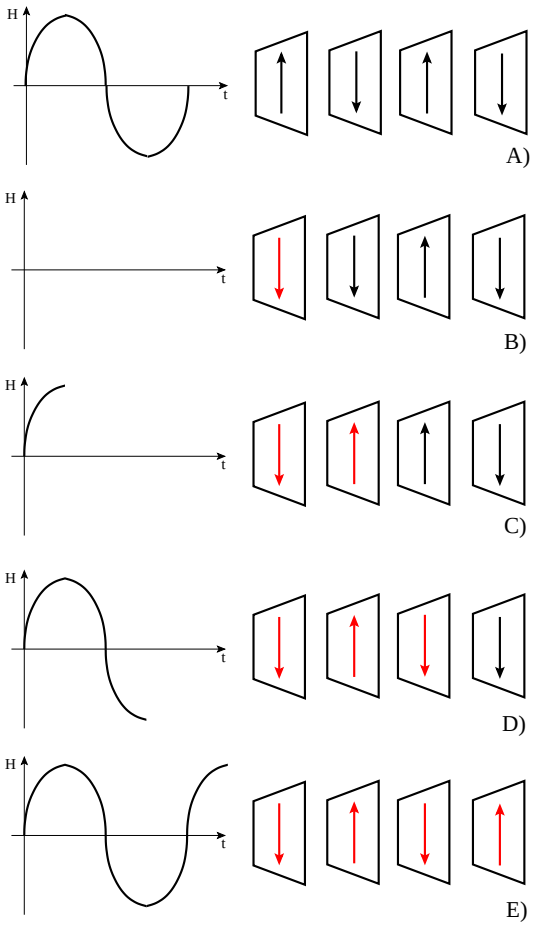



Figure 7.18. A) Global clock mechanism. B) At the beginning the input magnet change its state. C) When the magnetic field reach its maximum positive value, the sum of the clock magnetic field and the magnetic field generated by the input magnet, generates a magnetic field strong enough to switch the second magnet. D) When the field reach its maximum negative value the third magnet switches. E) The mechanism is repeated and all subsequent magnets switch with a domino effect following the global clock signal.

imposed by the clock signal. A second advantage of this clock system is that error probability is greatly reduced because magnets are not forced in the RESET state, which is an unstable state.

Unfortunately micromagnetic simulations show that this solution does not work. This is probably due to the lower control that there is in classic NML logic over local properties of magnets. In out-of-plane NML irradiating the magnets with ion beams it is possible to achieve a great local control over magnetic properties of the magnets. This kind of control cannot be achieved in classic NML simply

controlling the magnets shape. Since this clock mechanism have a great potential further investigation on this solutions are advised.

# Chapter 8

# Magnetoelastic clock

## 8.1 Magnetoelastic clock system

As described in Chapter 1 an external means, in the most classical case a magnetic field, is necessary to help the magnets to switch from one stable state to the other [22]. This magnetic field can be generated by a current flowing through a wire placed under the magnets plane (Figure 8.1.A). The resulting magnetic field is directed along the short side of the magnets, so when it is applied, magnets are forced in an intermediate unstable state with the magnetization vector rotated along the short side. When the magnetic field is removed magnets realign themselves following the input magnet. The clock frequency obtainable are in the range of 50MHz-1Ghz (see Chapter 7) [25][23][59], depending on the clocking technology chosen, so it is lower than the frequencies obtainable with CMOS.

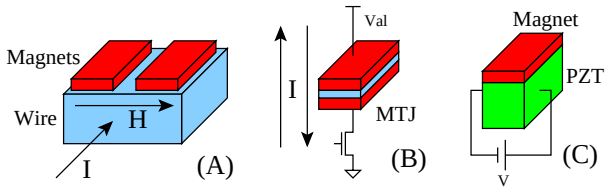



Figure 8.1. Proposed clock mechanism. A) A current which flows through a wire placed under the magnets plane generates the magnetic field that is used a clock signal. B) STT-current induced clocking for NML logic. MTJs junctions are used as basic elements and a current flowing through the magnets is used as clock. C) Multiferroic NML logic. The basic elements is a multilayered structure made by a piezoelectric material and a magnetic layer. This structure allows to electrically clock the dots.

It is important to remember at this point that the main interest beyond Nano-Magnet Logic is the expected very low power consumption, hundreds of times lower

than the expected power consumption of ultimate scaled down CMOS transistors [60]. Unfortunately this is true if only the energy required to switch the magnets is considered. If the losses in the clock generation system are considered this is no more true and all the advantages of this technology are wiped out. In [2] a current of 545mA in a copper wire 1um width is necessary to switch all the magnets, leading to a very high power consumption due to Joule losses. Moreover using this approach the local control of a clock zones is difficult to reach, because the magnetic field of one clock zone influences also the neighbors clock zones [61]. To solve this problem new clocking technologies were studied. An STT-current induced clock was proposed as a suitable way to reset the magnets (Figure 8.1.B) [23][40]. The basic element is no more a simple magnet, but is a Magnetic Tunnel Junction (MTJ), a multilayer structure composed by an insulator layer sandwiched between two magnetic layers. This is the same structure used in Magnetic RAM, and allows to reset every element with a current flowing "through" each element. The advantages of this approach are many: Much lower power consumption, built-in read/write system, perfect local control of each element and the possibility to use the well developed M-RAM technology. Another solution recently proposed uses multiferroic structures as base elements (Figure 8.1.C) [24][59]. The basic dots is composed by 40nm of piezoelectric material (PZT) and a 10nm magnetic layer. Every element is then controlled by applying a voltage of few mV. When the voltage is applied the strain of the magnetic layer, induced by the coupled piezoelectric material, makes the magnetization vector rotate toward the short side of the magnet, working as a reset mechanism. This system allows to reach the highest possible frequency with the lowest possible power consumption, with, at the same time, the possibility to use a voltage instead of a current to control the circuit.

While this approach is probably the best proposed solution for NML logic, and it will allows in the future to exploit the full potential of NML logic, it presents two major problems that makes the fabrication of the circuit quite difficult at the moment. The aspect ratio of every element is very low, with a difference of two nanometers between the two sides, and up to now it is difficult to reach a such low resolution also with Electron Beam Lithography. Moreover, this system requires to contact every elements with two electrodes to generate the required electric field, but electrodes of few nanometers are necessary and they are almost impossible to fabricate with current technology. As a consequence a different solution can be adopted, where the basic elements is a simple magnet and not a multiferroic structure. Magnets are deposited on a piezoelectric layer (PZT) driven by two parallel electrodes buried inside the PZT itself. While the performance obtained are lower than a pure multiferroic structure, they are remarkably better than all the other NML technologies, maintaining at the same time a strong link with the technological processes and the feasibility of the structure.

### 8.1.1 Clock structure

The basic idea is shown in Figure 8.2. A magnetic thin film is deposited above a piezoelectric substrate and it is patterned throughout lithography (Figure 8.2.A). When an electric field is applied to the substrate the piezoelectric material increases its length. If the thickness of the piezoelectric layer is much higher than the thickness of the magnetic layer, the strain in the piezoelectric substrate induces a strain of the same entity in the nanomagnets. The induced stress-anisotropy makes the magnetization vector rotates along the direction of the applied strain (Figure 8.2.B). This is the direct mapping of the clock principle that drives NanoMagnet Logic.

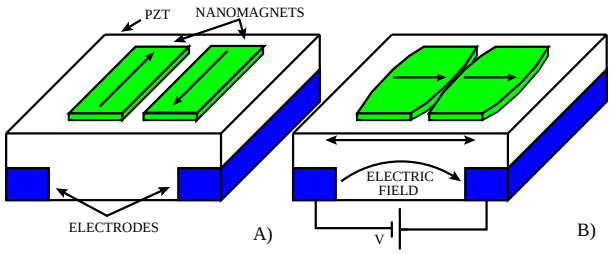



Figure 8.2. Magnetoelastic clock for NanoMagnet Logic. A) No voltage applied. B) Voltage applied to the PZT substrate. The strain induced in the nanomagnets change their magnetization.

It is a rather simple idea that was already demonstrated in [62]. In [62] an electric field was applied using two parallel electrodes placed on top and on the bottom of a piezoelectric (PZT - lead zirconate titanate) substrate. Relatively big (380x150 $nm^2$) Nickel magnets where successfully switched applying a small voltage (1.5 V). To apply the same concept for NML logic applications some issues arise. Electrodes placed on top of the PZT substrate are difficult to contact, because the surface of the PZT must be patterned with nanomagnets. Moreover with this configuration the electric field is applied perpendicularly while the strain is parallel to the PZT surface. In this way the strain and the electric field are coupled through the $d_{31}$ coefficient ($d$ coefficients, normally expressed in pm/V, describe the coupling between strain and electric field). The $d_{31}$ is much lower than the $d_{33}$ coefficient, that is used when the applied voltage and the strain are applied along the same direction. The solution here proposed uses electrodes embedded in the piezoelectric layer placed at both sides of the magnets, as a consequence electric field and strain lie along the same direction and they are therefore coupled through the $d_{33}$ coefficient. The consequence is that a lower voltage is required to generate the same strain and the power consumption is reduced. Moreover with this configuration electrodes can be contacted from the bottom, without interferences with the nanomagnets that are placed on top of the PZT layer.

### 8.1.2 Choice of magnetic material and magnet sizes

To choose the proper magnetic material and the nanomagnets geometry the maximum and minimum stress that can be applied must be evaluated. To evaluate the maximum stress first of all the maximum strain due to dielectric rigidity must be considered:

$$\xi_{MAX\_RIG} = Ef_{MAX} \cdot d \tag{8.1}$$

where $Ef_{MAX} = 20MV/m$ is the maximum electric field tolerated by the PZT layer, and $d = d_{33} = 150pm/V$ is the parameter that relates the absolute increase in length with the applied voltage. The previous value must be compared with the maximum strain achievable due to structural limitations:

$$\xi_{MAX} = min(\xi_{MAX\_RIG}, \xi_{MAX\_STRUCT}) \tag{8.2}$$

where $\xi_{MAX\_STRUCT} = 500 \cdot 10^{-6}$ [63]. Once the maximum strain is know it is possible to evaluate the maximum stress applicable to the magnets, making the assumption that the former are enough thin to make the PZT strain totally transferred on them:

$$\sigma_{MAX\_PIEZO} = Y_{Magnet} \cdot \xi_{MAX} \tag{8.3}$$

where $Y_{Magnet}$ is the Young module of the magnetic material chosen. But it is also necessary to consider the fracture stress of the magnet, which depends on the selected material. Consequently, the maximum stress that can be transferred to the magnets is:

$$\sigma_{MAX} = min\sigma_{MAX\_STRUCT}, \sigma_{MAX\_PZT\_COUPL} \tag{8.4}$$

The minimum stress is related to the height of the energy barrier between the two stable states, which depends on magnetic anisotropy. There are two types of magnetic anisotropy that must be considered: Magnetocrystalline anisotropy and shape anisotropy. Magnetocrystalline anisotropy, which is related to the structure of the crystal, leads to a very high energy barrier. As a consequence the maximum applicable stress is not strong enough to rotate the magnetization vector. To be a suitable candidate as NML logic the magnetic material must have a negligible value of magnetocrystalline anisotropy. Shape anisotropy is related to magnets shape: If magnets have an aspect ratio different from 1, at the equilibrium magnetization will lie along the longer side of the magnets. In this case the height of the energy barrier between the two stable states depends on the aspect ratio of the magnets. The minimum applicable stress is therefore the stress that generates a stress anisotropy at least equal to the shape anisotropy:

$$\frac{1}{2}\mu_0 N_d M_s^2 V = \frac{3}{2}\lambda_s \sigma V \tag{8.5}$$

where $N_d$ is the demagnetization factor, $M_s$ is the saturation magnetization, $V$ is the volume and $\lambda_s$ is the magnetostrictive coefficient. The minimum applicable stress is therefore:

$$\sigma_{MIN} = \frac{\mu_0 N_d M_s^2}{3\lambda_s} \tag{8.6}$$

NML logic requires the use of single domain nanomagnets, that means with sides lower than 100nm. In literature magnets are normally 50x100 $nm^2$ [41] width or 60x90 $nm^2$ [2] width. A value of 50 nm was therefore chosen for the shortest side of the magnets, while the thickness is around 10nm. The magnets aspect ratio determines the value of the shape anisotropy, i.e. the height of the energy barrier. To have a reasonably small value of error probability ($p < e^{-30} \approx 10^{-13}$), the energy barrier at room temperature must be at least

$$\Delta E = 30 K_b T \approx 1.24 \cdot 10^{-19} J \tag{8.7}$$

that leads to a minimum aspect ratio of 1.06, which means a minimum sizes for the magnets of 50x53x10 $nm^3$. To choose a suitable magnetic material the minimum stress necessary to reset the magnets was evaluated starting from an aspect ratio of 1.06 to 2, comparing this value to the maximum applicable stress. Results are shown in Figure 8.3. For most classical magnetic materials, like Iron or Cobalt, there is no range in which the circuit can work properly. Figure 8.3.A shows the results obtained for Iron, the minimum required stress is always bigger than the maximum applicable stress. This is nothing of strange since Iron is a material with negligible magnetostriction. Figure 8.3.B shows the results obtained for Nickel. As it is possible to see there is a range in which the device can operate, from 1.06 to 1.15 aspect ratio (53-57.5 nm). Things change dramatically if a high magnetostrictive material, like the Terfenol is considered (Figure 8.3.C). As it is possible to see the working range increases a lot, from 1.06 to 1.57 aspect ratio (53-78.5 nm). Moreover the required stress is lower than the required stress for the Nickel (60 MPa for Nickel, 28 MPa for Terfenol).

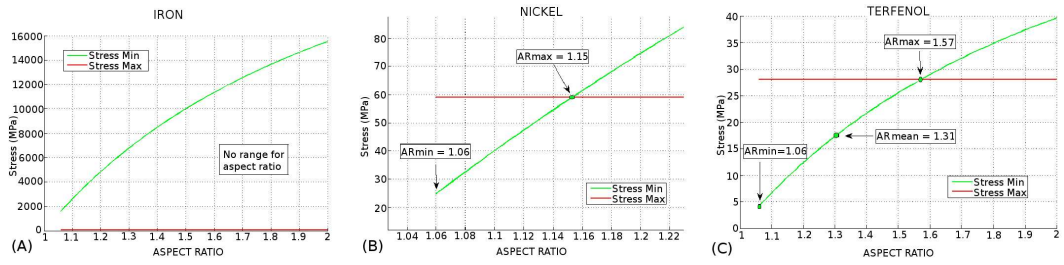



Figure 8.3. Comparison between the minimum required stress and the maximum applicable stress for different magnetic materials. A) Iron. B) Nickel. C) Terfenol.

Although both Nickel and Terfenol can be suitable targets for this technology, the limited operative range of Nickel can be a problem if process variations are considered. For example considering a process variation of +/-10% things changes, as it is possible to see from Figure 8.4. The lower and upper curves represent the minimum required stress considering a process variation of -10% (lower curve) and +10% (higher curve). The chosen value of aspect ratio (on the central curve) shifts up or down (on one of the lower or upper curves or in some middle point) if, due to process variations, the aspect ratio of the magnets changes. As a consequence the aspect ratio must be chosen in a way that, in case of random shifting due to process variations, it still falls in the acceptable range (between 0 and the maximum applicable stress). Figure 8.4.A shows the working range of Nickel considering process variations of +/-10%. As it is possible to see there is no point that lies in the operative range. An aspect ratio of 1.17 can balance negative variations but it doesn't work with positive variations. This means that Nickel is very sensitive to process variations, it cannot tolerate variations bigger than 5-7%. Figure 8.4.B shows instead the working range for Terfenol. As it can be noted considering process variations the minimum value for the aspect ratio become 1.18 while the maximum become 1.42. This means that Terfenol, has a very good working range and can tolerate process variations also near +/-20%. It is clear from these analysis that high magnetostriction materials, like Terfenol, are perfect candidates for this application.

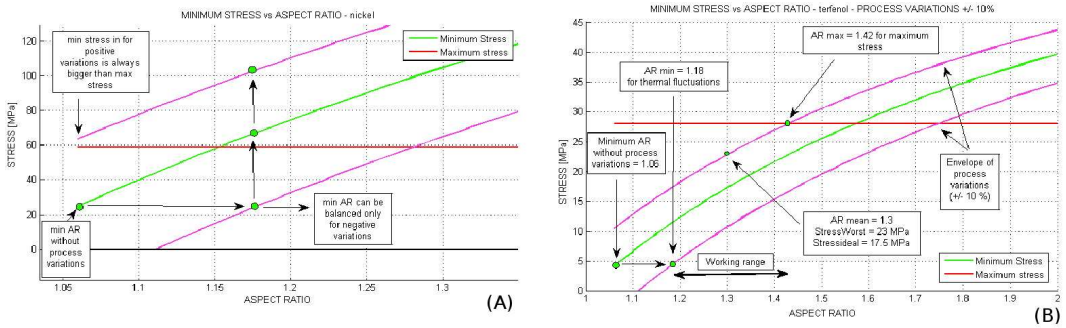



Figure 8.4. Working area of different magnetic materials considering process variations. A) Nickel. B) Terfenol.

From this analysis the material chosen is Terfenol, with sizes of 50x65x10 $nm^3$. Comparing this geometry with the one proposed in [24] the difference between the smaller and bigger magnet is higher (15nm instead of 2nm) and magnets are simple single layer structures, this means that they are easy to fabricate and also tolerant to process variations.

### 8.1.3 Circuit Layout

The layout of the circuit must take into account two important problems: Signal propagation and fabrication processes. The solution here proposed is shown in Figure 8.5. Parallel electrodes are buried under a PZT layer, and nanomagnets are deposited directly on top of it. This solution is technology-friendly because is compatible with CMOS planar technology and, supposing to have a high-end resolution lithographic system, can be already fabricated. After the deposition of metal (Platinum) to create the electrodes, the PZT is deposited on top of them. Nanomagnets can be fabricated by depositing a thin film of magnetic material on top of it and then patterning the film using lithography.

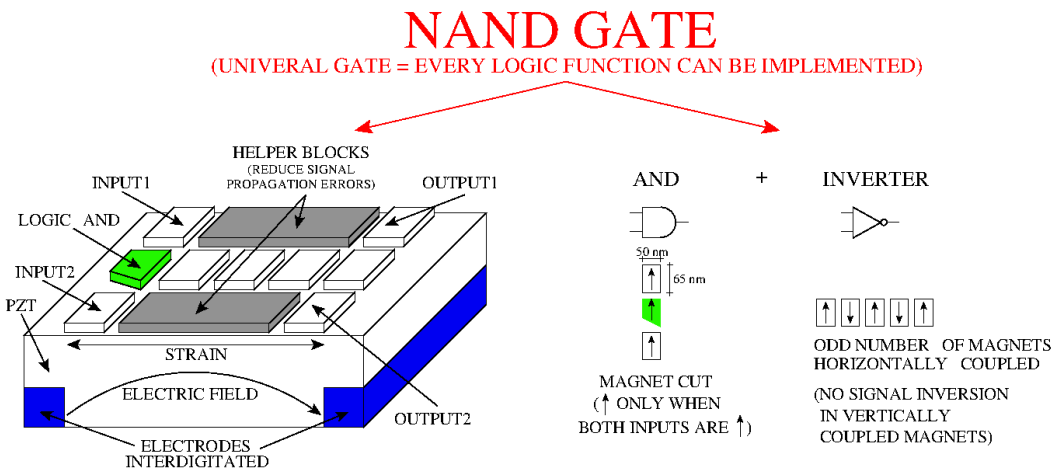



Figure 8.5. Proposed magnetoelastic clock system. Parallel electrodes buried under the PZT layer generate the electric field. The strain transfers to the magnets that are reset. Input and output propagate vertically from each corner. Shielding blocks are used to avoid propagation errors.

The fabrication process is relatively simple but the problem is how the electric field will be distributed in the piezoelectric layer. Figure 8.6 shows a Comsol Multiphysics [58] Simulation of the structure, which shows the distribution of the electric field. The applied voltage is 1V and an electric field of 3-4 MV/m is generated almost uniformly between the two electrodes. In correspondence of the electrodes the electric field decreases abruptly and reach a value of about 2MV/m near the borders. The strain of the PZT is proportional to the electric field, so it is clear that the strain will be lower near the areas corresponding to the electrodes. However, due to mechanical continuity, the higher strain of the central area will induce a strain in the neighbor areas, also in the area exactly above the electrodes where the electric field has a very low value. The situation will improve reducing the distance between the electrodes and the PZT surface, however from technological point of view it

is more complex to fabricate. Basing on the results of Figure 8.6 it is possible to approximate the strain as uniformly applied in the area between the two electrodes.

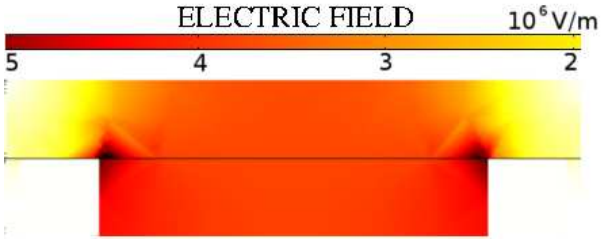



Figure 8.6. Comsol Multiphysics simulation of the structure. The electric field (and as a consequence the strain) is almost uniform between the two electrodes.

The consequence is that to obtain working circuits magnets must not be placed in the area correspondent to the electrodes. The design is therefore based on 2 input AND/OR gates [6], as shown in Figure 8.5 AND/OR gates are made by three magnets, the shape of the central magnet is changed to obtained the desired logic function. The advantage of this solution is that inputs come from up and down directions, where there are no electrodes. Another point is that, in NML logic the horizontal coupling is antiferromagnetic, every magnet has the inverted value of its predecessor. So, if the number of magnets in the clock zone is odd the signal is inverted. Placing therefore and AND/OR gate in a clock zone with a width equal to an odd number of elements generate a universal NAND/NOR gate that can be used as basic block to build any circuit.

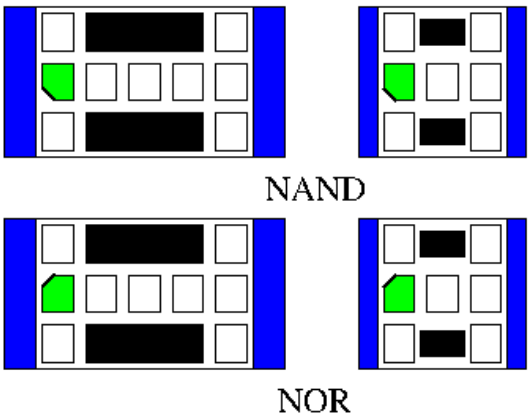



Figure 8.7. Universal NAND/NOR gates. Every gate is high 3 magnets and with a variable width of 3 or 5 magnets.

Ideally the width of the clock zone must be equal to one magnet to obtain the

maximum possible clock frequency, as shown in [24]. However this approach has two disadvantages: It increases the latency of the circuit and it make the fabrication of the structure and the signal propagation almost impossible. Increasing the latency of the circuit reduce the throughput in presence of sequential circuits [42]. Moreover, the distance between the electrodes will be smaller and the whole structure more difficult to fabricate. Also, since magnets cannot be placed over the area of the electrodes, with a width of one magnet there is not enough space to propagate the output signal of the logic gate. The width of the gate chosen is 3 or 5 magnets, as shown in Figure 8.7. Inputs comes from up-left and bottom-left corners, output of the AND/OR gate is propagated to the up-right and down-right corners. In this way signals can propagate to the others parts of the circuit avoiding the area of the electrodes. Helper blocks [3] are used to help the signal propagation and reduce the error probability. With a width of 5 magnets the critical path (the maximum number of magnets between input and output) is higher, 7 magnets instead of 5 magnets in case of a width equal to 3 elements. Since the clock frequency depends on the critical path, with a width of 5 magnets the clock frequency will be lower but the structure is bigger and easier to fabricate. Bigger sizes are not possible, because, no only the clock frequency will be much lower, but the length of the critical path will be too big, increasing the error probability during the magnets switching.

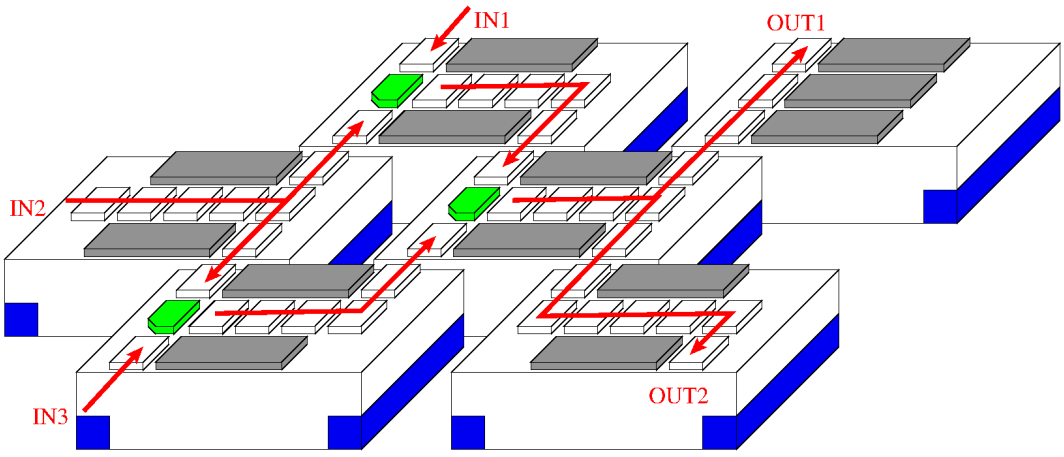



Figure 8.8. Circuit layout. Each row is composed by many clock zones of area 3x3 or 3x5 magnets. Alternate rows are shifted to allows signal propagation.

The circuit layout is shown in Figure 8.8. Clock zones are made by mechanically isolated cells of 3x5 or 3x3 magnets. Every cell is an independently actuated clock zone, where logic gates or interconnection wires can be placed. To create this layout it is possible to pattern the PZT substrate, removing the PZT (Figure 8.9) [64][65]. It is possible to dig through the PZT until the bottom, or to remove only a part of

the PZT to mechanically isolate the areas. In both solutions a perfect mechanical isolation is obtained, but probably the complete removal of the PZT will reduce parasitic parameters. Clearly the resolution of the optical lithography must be quite high to remove only a small area of the piezoelectric layer. Theoretically would be sufficient to remove few nanometers between the clock zones, but it is quite difficult to obtain with current lithographic processes.

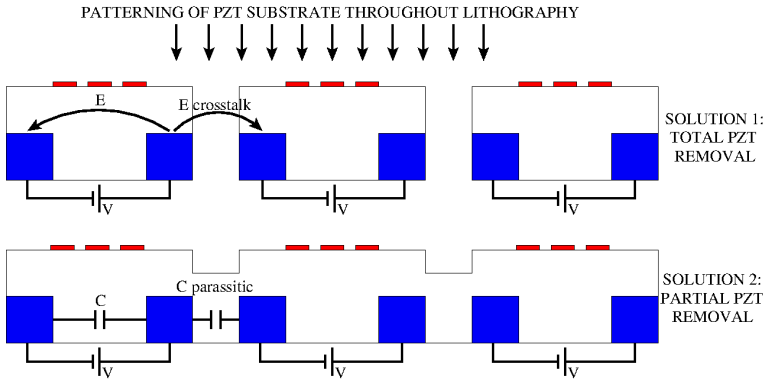



Figure 8.9. PZT can be patterned to obtain mechanically isolated cells. Two solutions are possible: Complete or partial removal of the PZT.

Signal propagation happens through the corner of each clock zone, to avoid the area of the electrodes. To allow this, there must be a shifting in each row of clock zones, as can be seen from Figure 8.8. As it is possible to see from Figure 8.8, with this layout the width of the clock zone must therefore be chosen according to the size of the electrodes. With a width of 3 magnets, electrodes must be 30-40 nm width. while with clock zones 5 magnets width electrodes can be approximately 70-100 nm width, a size that is available in 22 nm CMOS technology. Since this approach is based on universal NAND/NOR gates every circuit can be implemented, moreover the circuit layout is quite regular, and this always helps the technological fabrication.

### 8.1.4 Performance analysis

To verify the effectiveness of the solution proposed in this work, circuit performance were estimated both in terms of timing and power consumption. Figure 8.10 shows the timing characteristics obtained through Magpar [66] simulations. Magpar is a finite element simulator based on Landau-Lifshitz-Gilbert equation. In Figure 8.10.A the time required to reset the magnets is indicated. About 1 ns is necessary to complete reset the magnets. From Figure 8.10.B it is possible to see that also the switch time of every magnets is near 1 ns. The clock frequency can therefore be

estimated starting from this data. The clock period must last enough to allow the reset of the magnets and their successive realignment. So, as a first approximation the minimum clock period can be calculated as:

$$T_{ck} = T_{RESET} + N{*}T_{SWITCH} \tag{8.8}$$

where $N$ is the number of magnets in the critical path (5 considering a 3x3 NAND, 7 considering a 5x5 NAND). However the situation is more complex, because, in a chain of magnets, one element start to switch before its neighbor has reached a stable state. So the clock period is not directly the sum of N switching times As a consequence the maximum clock frequency obtainable is around 200MHz for 3x3 NAND/NOR gates and 150MHz for 3x5 NAND/NOR gates. The frequency is lower than the one obtained in [24], but this is due to the higher number of element in the critical path.

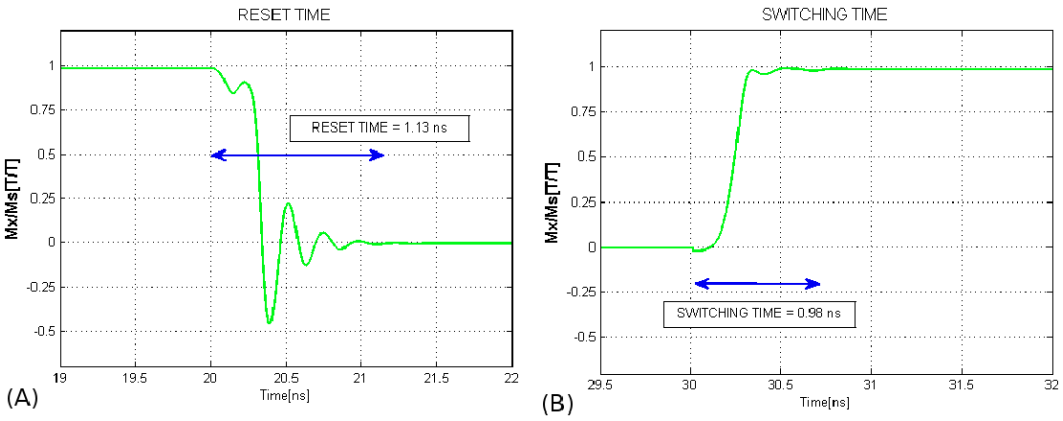
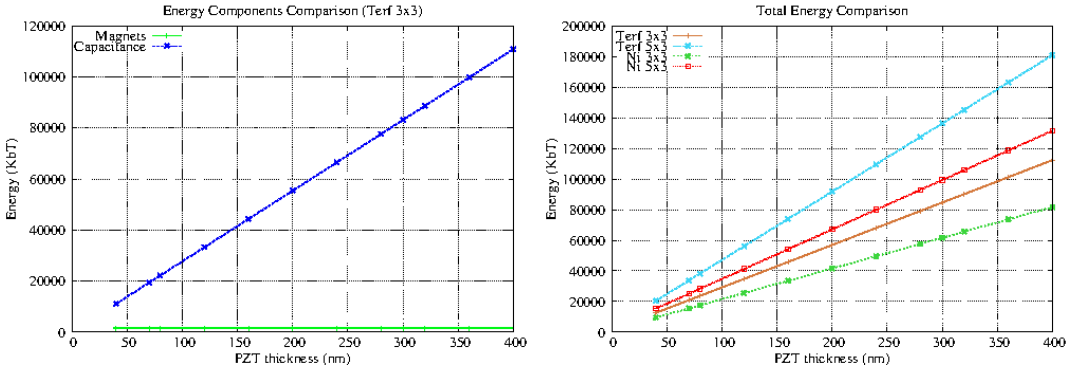



Figure 8.10. A) Nanomagnets RESET time. B) Nanomagnets SWITCH time. Both times are in the order of 1ns.

However speed is not the major advantage of NML logic. This technology is studied for the low power consumption obtainable. There are two main sources of power consumption, the energy required to force the magnets in the RESET state and the losses in the clock generation system. As explained in Chapter 7 the energy required to RESET a magnet is about $180K_bT$. The origin of this lie in the fact that an abrupt switching was applied to achieve the maximum circuit speed. Using an adiabatic switching (i.e. very slow rise and fall time for the clock signals, in the order of many nanoseconds) this energy can be reduced to $30K_bT$, greatly reducing the obtainable circuit speed. However the major source of power consumption in a NML circuit are the losses in the clock generation system, and compared to this source also the value of $180K_bT$ is negligible. As a consequence an abrupt switching

was used to maximize the clock frequency, which lead to a power consumption of $181K_bT$ for each magnet.

The second and much more important source of power consumption are the losses in the clock generation system. First of all there is a contribution due to the Joule losses in the transmission wires. This can be an important source of power consumption in current-based clock systems, but not in this case. Every NAND/NOR gate is a capacitor, where the contact are the armors and the PZT is the dielectric. Since the PZT is an insulator and the voltage applied is quite low (lower than 1V) the current that flows through the circuit is almost zero and the joule losses are negligible. The exact value is not reported because the energy lost due to Joule losses is twelve orders lower than the energy lost for magnets reset. Since the whole structure is a capacitor the biggest source of losses, is the energy necessary to charge the capacitor $\frac{1}{2} \cdot C \cdot V^2$. Since the capacitance of the proposed structure can be quite big, the impact can be high on the global energy consumption, as can be seen from Figure 8.11.




(a) (b)

Figure 8.11. a) Comparison between energy consumption components for a 3x3 NAND/NOR with magnet of Terfenol. Energy required to reset the magnets is constant and much lower than energy lost to charge the capacitor. b) Comparison between NAND/NOR with different sizes and different materials. Nickel has a lower energy consumption due to a higher Young modulus.

Figure 8.11.a shows the comparison between the energy required to reset the magnets of a 3x3 NAND/NOR gate with magnets of Terfenol, with different thickness of the PZT layer (from 40nm to 400nm). The capacitance of the structure strongly depends on the PZT thickness, so the energy consumption necessary to charge the capacitor increases linearly with the thickness of the piezoelectric layer. However, as can be noted from Figure 8.11.a, the difference between the capacitance energy and reset energy is very high also with the smallest PZT thickness of 40nm.

To reset 7 magnets that compose the gate a total energy of $1400K_bT$ is necessary, while $11000K_bT$ of energy is lost due to the capacitance in case of 40nm of PZT thickness. As a consequence there is a difference of 10 times also in the best case, but the gap increases dramatically with the PZT thickness. The increment of energy consumption with the thickness of the piezoelectric layer can be a problem, because it is difficult to obtain a PZT thin film with a very small thickness.

Figure 8.11.b shows instead a comparison of the total energy consumption between NAND/NOR gates with sizes of 3x3 magnets and 3x5 magnets using Terfenol and Nickel as magnetic material. The behavior is the same as shown in Figure 8.11.a, the energy consumption increases linearly with the PZT thickness. Moreover increasing the size of the logic gate the energy consumption also increases. Increasing the width of the gate increases the distance between the electrodes. therefore to have the same electric field a bigger voltage is necessary. But most importantly the energy consumption changes with the material, particularly using Nickel instead of Terfenol the energy is lower. This can be explained considering the relation between strain and stress in a material. High magnetostriction materials, like Terfenol, have a bigger change in the magnetization with the same applied stress with respect to low magnetostriction materials like Nickel. However stress and strain are bonded through the Young modulus. The Young modulus of Nickel is more than 2 times bigger than the Young modulus of Terfenol. This means that Nickel requires a bigger stress to change its magnetization, but to generate that stress a lower strain (and therefore voltage) is required. The energy consumption is therefore lower in case of Nickel. However Terfenol remains the preferred choice because it allows to use magnets with an higher aspect ratio and has a better tolerance to process variations.

Table 8.1. Power comparison among the main NML implementations.

| | Energy (fJ) | Clock (MHz) |
|---|---|---|
| Magnetic Field | 62 | 50-100 |
| STT-current | 11 | 100-200 |
| Multiferroic | 0.004 | 500 |
| Magnetoelastic without compensation | 0.052 | 200 |
| Magnetoelastic with compensation | 0.006 | 200 |
| CMOS LOP 22nm | 0.110 | - |

Finally a comparison between the different clock systems is mandatory. Table 8.1 shows the total energy consumption and the obtainable frequencies for a NAND gate made with the main NML logic implementations. For the classic NML magnetic field driven a energy consumption of $30K_bT$ for each magnet is considered. Moreover

the energy losses due to Joule effect was also estimated. The wire has a section of about 400x400nm and a length of about 200nm, is made of copper and the current value is 2mA (extrapolated from [2]). This lead to an energy consumption of 62fJ for a NAND gate. The frequency achievable, due to the use of adiabatic switching, is in the range of 50-100MHz [25]. For STT-current induced clock data are obtained from [23]. An energy of 1,6fJ is necessary to reset the magnets, that gives a total of 11fJ for a NAND gate. This system is far better than the magnetic-field based clock. Frequencies obtainable are in the range of 100-200MHz [40].

Considering instead multiferroic logic, data shown in [24] indicates a total energy required to operate a NAND gate of about 4 aJ, at least 3-orders better than the current based approaches. The frequencies is also relatively high, at about 500MHz. With the solution proposed here an energy consumption of 52aJ and a maximum frequency of 200MHz were obtained. Although these values are far better than the current based approaches a pure multiferroic logic still shows better performances. However two things can be noted: First this approach is technology-friendly and circuits could be already fabricated with modern fabrication processes, second the energy consumption can be greatly reduced. The reduction of energy consumption can be obtained because the main source of losses is the energy required to charge the capacitance. Setting up a LC resonant circuit allows to compensate this energy, so that the energy required by the capacitor can be given by the inductor. The advantages can be clearly understood from table 8.1, with compensation the energy consumption drop of 10 times. Clearly in a real case there will be always a resistance so it is impossible to completely recover the energy lost in the capacitor, but this component of losses can be at least greatly reduced. This is important particularly in case a PZT with higher thickness will be required, allowing to obtain a low energy consumption also with bigger structures. A final interesting thing can be observed from table 8.1, where the energy consumption of a 22nm Low Operating Power CMOS NAND gate obtained from the ITRS Roadmap [67] is shown. The energy consumption is double than this NML solution, but with energy compensation the gain increase to 20 times. This further underline the big advantage of electrically clocked NML circuit over CMOS technology. Clock frequency of CMOS is not reported in table 8.1 because is much more high than NML, which is a technology studied for its low power consumption and not for its speed.

## 8.2 Magnetoelastic clock system fabrication

The main goal is to experimentally validate the proposed clock solution, however the lithography processes available in our facilities do not reach the necessary resolution. As a consequence we are building a more simple structure that will allow us to demonstrate this clock mechanism. The structure of the demonstrator is shown in

Figure 8.12. Two interdigitated electrodes are buried under a PZT layer. A NML wire, a simple chain of magnets, is used as test circuit. Magnets are located in the area between two electrodes arms. The aim of this structure is to demonstrate that, when a voltage is applied to contact pads, magnets are forced in the RESET state. Removing the voltage, magnets should align antiferromagnetically, demonstrating therefore the correctness of this clock solution.

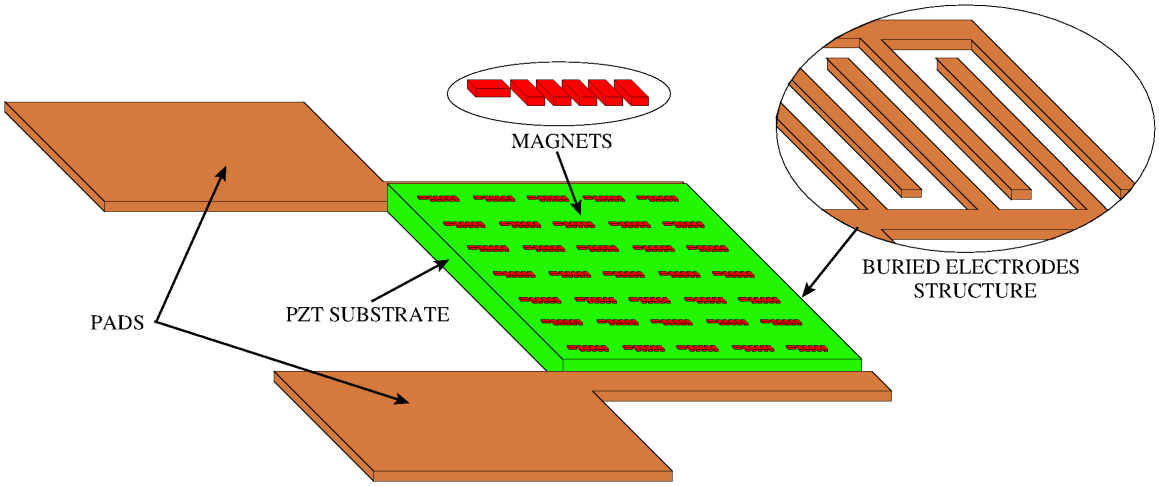



Figure 8.12. Structure of the proposed circuit demonstrator. Two interdigitated electrodes are covered by a PZT layer. Magnets are located in the area between two electrodes arms. Contact pads are used to apply the voltage to the structure.

The fabrication process of the demonstrator is shown in Figure 8.13. First a photoresist is deposited on the substrate (Figure 8.13.B). Then the photoresist is patterned through direct laser writing lithography, the metal is deposited over the photoresist which is successively removed, leaving only the electrodes (Figure 8.13.C). In the next step the PZT layer is created on the electrodes structure through spin coating (Figure 8.13.D). The contact pads area must be then cleared from the PZT to allow the contact with external instruments to apply the voltage (Figure 8.13.E). A magnetic film is deposited on the existing structure (Figure 8.13.F) through sputtering. Also contact pads will be covered with the magnetic material but this is not a problem since it is an electric conductor. Finally the magnetic circuit is patterned using electron beam lithography (EBL) or Focused Ion Beam (FIB) lithography.

### 8.2.1 Electrodes

Due to technical limitations we have not yet obtained the complete structure, however in Figure 8.14 some preliminary results are shown. Figure 8.14.a shows a

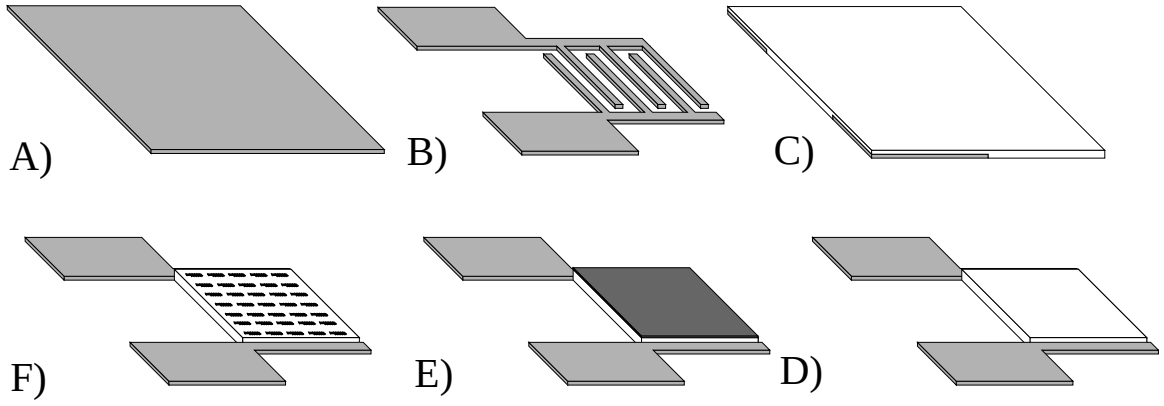



Figure 8.13. Fabrication Process. A) Metal deposition. B) Electrodes patterning through IDE lithography with laser writer. C) Deposition of PZT trough spin coating. D) PZT removal from pads area. E) Deposition of magnetic material through sputtering. F) Patterning of magnets through EBL or FIB lithography.

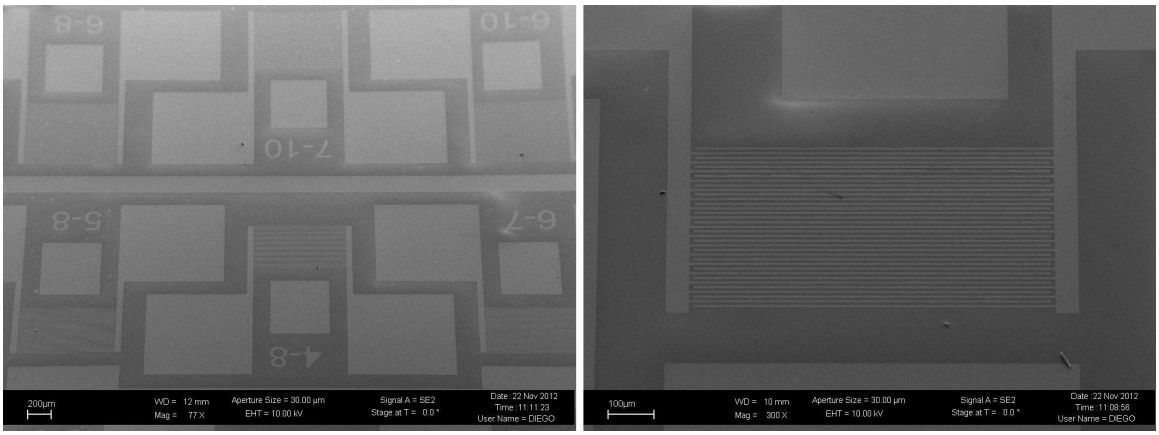
scanning electron microscope (SEM) image of the electrodes structure. It is possible to observe the contact pads and the electrodes. Arms of the two electrodes are alternatively interleaved, in this way there will be a strain between each couple of arms. The maximum resolution that we can obtain with our lithography process is 2um, so, as it is possible to note from Figure 8.14.b, electrodes sizes are in the micrometer range. This sizes are much bigger than desired, however they are enough if the purpose is the demonstration of the effective magnets reset and switching.

### 8.2.2 PZT substrate

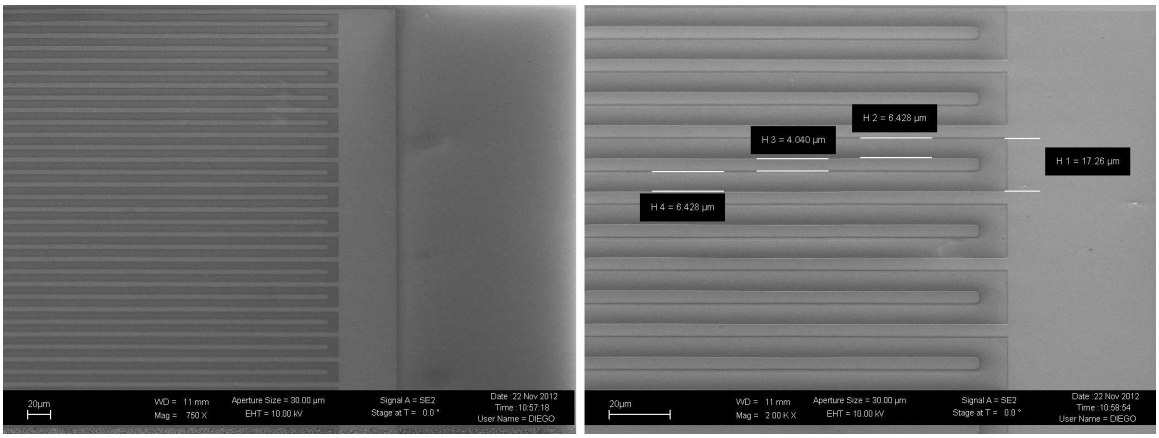
Figure 8.15 shows instead a SEM image of a typical PZT substrate [68]. It is possible to observe its typical grain structure, with grain sizes in the range of a hundred nanometers. The average roughness is quite low, 3nm, so it should be possible to directly deposit the magnetic material on top of the PZT. Normally an interface layer is used between the PZT and the magnets [62], however this will reduce the mechanical coupling so we are trying to avoid it. We can obtain PZT film with a thickness in the range of 70-600 nm. The obtained value of $d_{33}$ coefficient for a 600nm film is around 200 pm/V.

### 8.2.3 Magnetic materials

Figure 8.16 shows instead a film of Iron-Terbium. Iron-Terbium is a material similar to Terfenol with high magnetostriction that we are studying since [69]. As it is possible to see its surface is pretty rough and the thickness is quite high, 500nm. Our




(a)




(b)

Figure 8.14. Detail of the electrodes structure. Sizes are in the range of micrometers because the resolution limit of our lithography process is 2um.

efforts are now concentrated on the creation of thinner films with a lower roughness and its deposition on the PZT layer. After a good film is obtained we will focus on the magnets patterning completing therefore the demonstrator.

### 8.2.4 Magnetic dots

To fabricate magnets the best solution would be the use of Electron Beam Lithography, which theoretically allows to obtain very small structures without damaging the substrate. Unfortunately our effort with EBL lithography has not given good results, probably due to the influence of the substrate, which is a strong insulator. Thanks to the insulating nature of PZT the electron beam is not well conducted

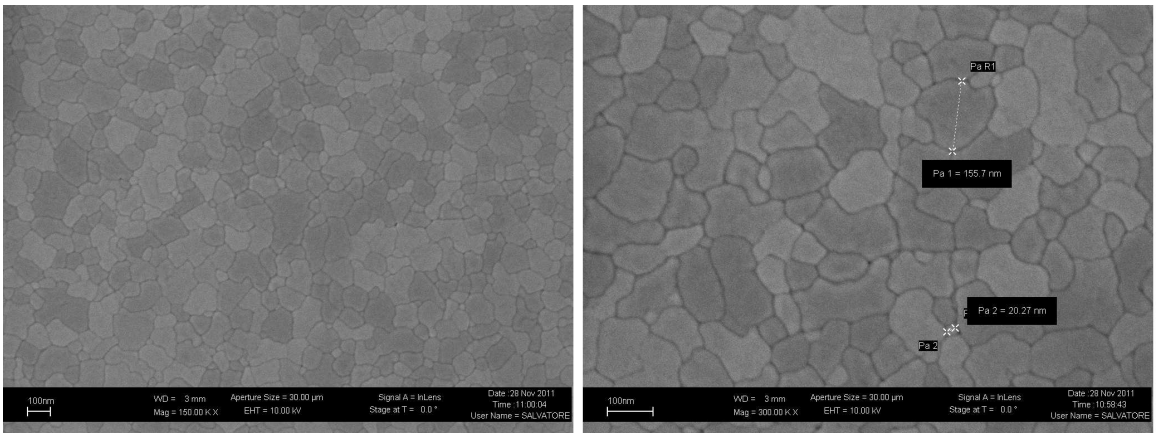



Figure 8.15.

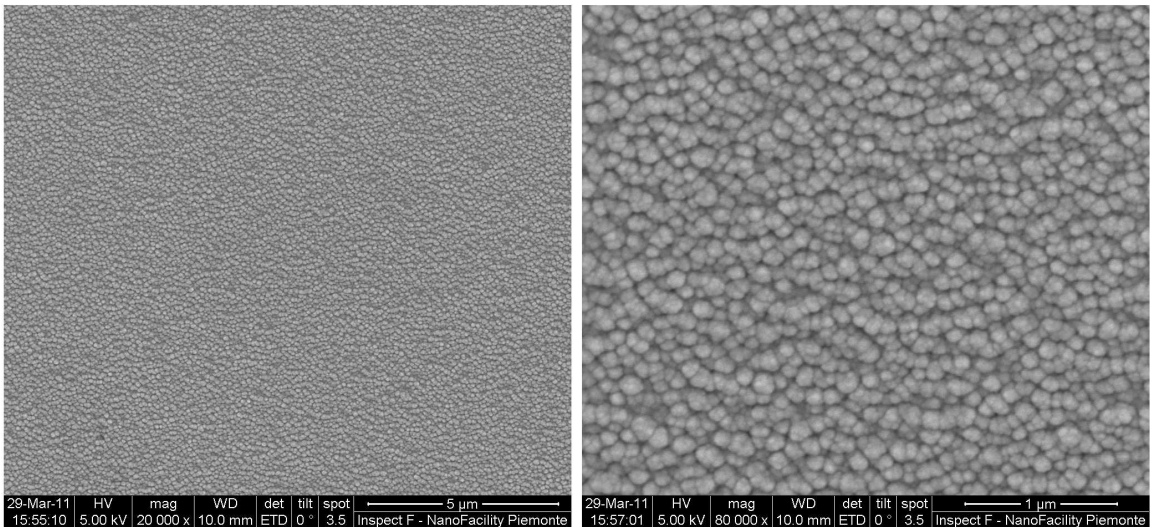



Figure 8.16.

through the substrate, greatly reducing the effective resolution of the lithographic process. Moreover the acceleration voltage is limited to only 30 kV, and, as a consequence, probably, it is not possible to avoid the proximity effect in such small structures. The proximity effect is generated by the electrons that do not penetrate through the substrate, and are therefore bounced back, damaging the material nearby.

To overcome this problem a Focused Ion Beam lithography (FIB) was used instead of EBL. The results obtained are relatively good, since it is possible to fabricate

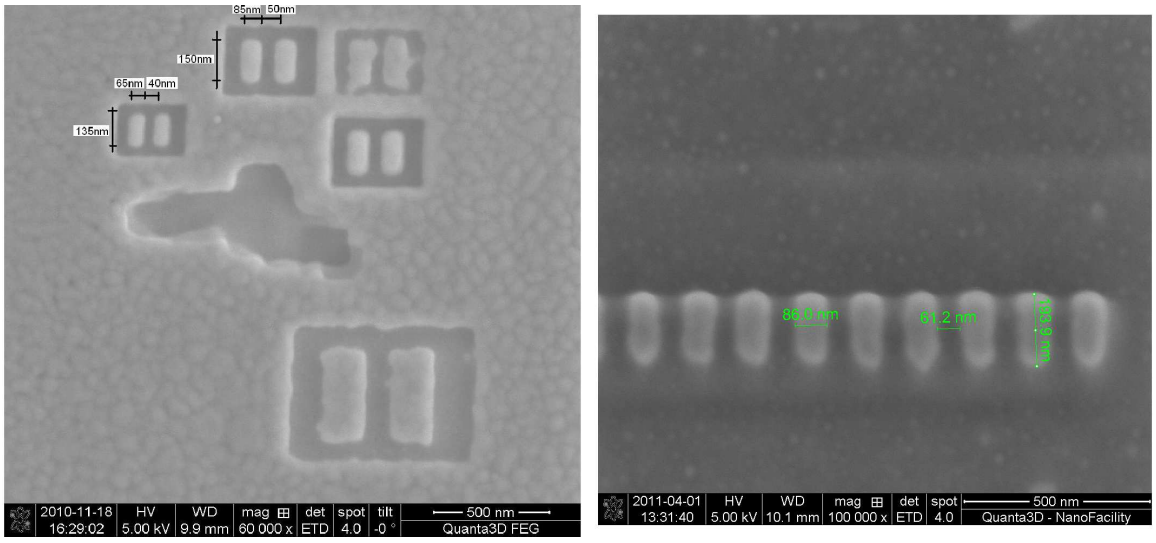



Figure 8.17.

magnets with a resolution up to 20-30 nm. Figure 8.17 shows first preliminary results of magnets fabrication through FIB lithography, where, firstly simple magnets and then more complex structures like wires are were successfully fabricated. The smallest magnets that were successfully fabricated are 70x140 $nm^2$ as can be seen from Figure 8.17.

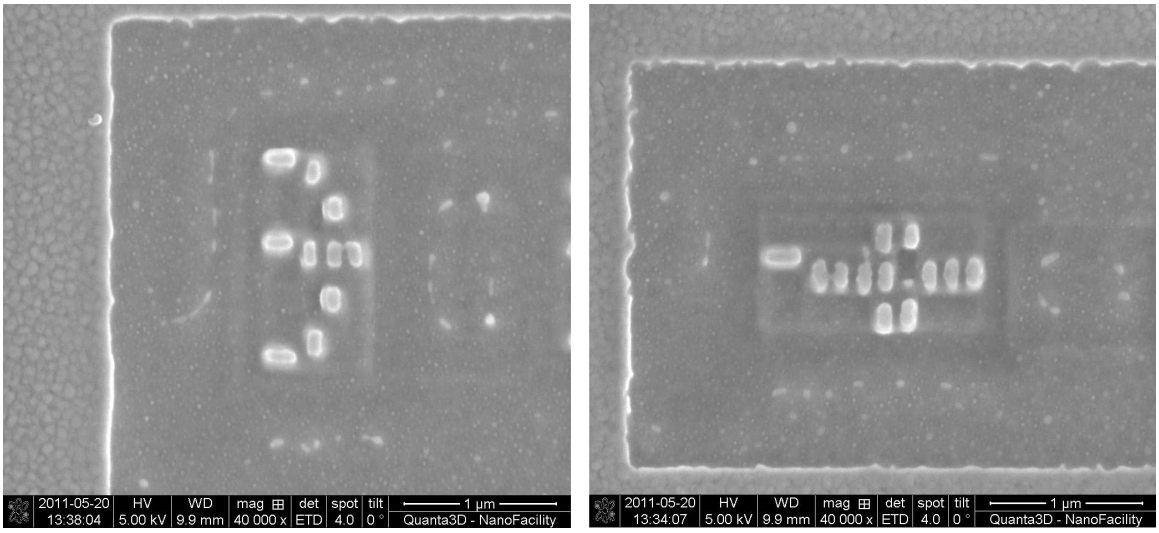



Figure 8.18.

The fabrication process was then improved, allowing to obtain complex structures improving at the same time the quality of the obtained magnets. Figure 8.18 shows an example of the results obtained, where a majority voter and an inverter are shown. The disadvantage of the FIB lithography is that it causes damage to the substrate. It is not possible to remove only the magnetic material but also others 50-60 nm of PZT are removed. This can dramatically change the properties of the piezoelectric layer. Since now we were not able to obtain a measurement through Magnetic Force Microscope (MFM) of the fabricated dots. This can be caused by the lack of resolution of the machine used or the influence of FIB lithography with change the magnetic properties of the magnetic material, by means of the high heat generated by the ion beam. We are currently working to understand the reasons behind our inability to measure the magnetization of the fabricated dots, trying at the same time to improve the fabrication process.

## 8.3 Acknowledgment

I thank Nanofacility Piemonte and Compagnia di San Paolo for the support. I would also like to thank ChiLab laboratory (Materials and Processes for Micro & Nano Technologies - Chivasso) for the technical support in PZT deposition.

# Part III

# Appendix

# Appendix A

# Publications

## A.1 Conferences

- Marco Vacca, Mariagrazia Graziano and Maurizio Zamboni **"Magnetic QCA: A full magnetic Logic"**, Magnet2011, Italian conference on magnetism, 23rd-25th february 2011.

- Marco Vacca, Davide Vighetti, Matteo Mascarino, Luca Gaetano Amaru, Mariagrazia Graziano and Maurizio Zamboni **"Magnetic QCA Majority Voter Feasibility Analysis"**, Prime 2011, 3rd-7th July 2011, DOI 10.1109/PRIME.2011.5966275.

- Marco Vacca, Mariagrazia Graziano, Danilo Demarchi and Gianluca Piccinini **"TAMTAMS: a flexible and open tool for UDSM process-to-system design space exploration"**, ULIS 2012, Ultimate integration on silicon, 5th-7th March 2012, DOI 10.1109/ULIS.2012.6193377

- Marco Vacca, Stefano Frache, Mariagrazia Graziano and Maurizio Zamboni **"ToPoliNano: A synthesis and simulation tool for NML circuits."**, IEEE Nano 2012, International conference on Nanotechnology, Accepted publication, 20th-23th August 2012.

- Marco Vacca, Giovanna Turvani, Fabrizio Riente, Mariagrazia Graziano, Danilo Demarchi and Gianluca Piccinini **"TAMTAMS: An Open Tool to Understand Nanoelectronics"**, IEEE Nano 2012, International conference on Nanotechnology, Accepted publication, 20th-23th August 2012.

- Muhammad Awais, Marco Vacca, Mariagrazia Graziano and Guido Masera **"FFT implementation using QCA"**, ICECS 2012, IEEE International Conference on Electronics, Circuits, and Systems, 9th-12th December 2012.

- Gianvito Urgese, Mariagrazia Graziano, Marco Vacca, Stefano Frache and Maurizio Zamboni **"Protein Alignment HW/SW Optimizations"**, ICECS 2012, IEEE International Conference on Electronics, Circuits, and Systems, 9th-12th December 2012.

- Marco Vacca, Mariagrazia Graziano and Maurizio Zamboni **"Nano-magnet Logic: An Architectural Viewpoint"**, 2013 Workshop on Field-Coupled Nanocomputing, Tampa (Florida), 7th-8th February 2013.

- Marco Vacca, Luca Di Crescenzo, Mariagrazia Graziano, Maurizio Zamboni, Alessandro Chiolerio, Andrea Lamberti, Emanuele Enrico, Federica Celegato, Paola Tiberto, Luca Boarino **"Electric clock for Nano-magnet Logic Circuits"**, 2013 Workshop on Field-Coupled Nanocomputing, Tampa (Florida), 7th-8th February 2013.

- Marco Vacca, Stefano Frache, Mariagrazia Graziano and Maurizio Zamboni **"ToPoliNano: Nano-magnet Logic Circuits Design and Simulation"**, 2013 Workshop on Field-Coupled Nanocomputing, Tampa (Florida), 7th-8th February 2013.

- Marco Vacca, Massimo Ruo Roch, Guido Masera, Giacomo Frulla, Piero Gili **"WindDesigner: An Open Tool for Analysis and Design of Wind Generators"**, International Conference on Clean Electrical Power (ICCEP) 2013, Alghero (Sardegna), 11th-13th June 2013, accepted publication.

## A.2 Journals

- Mariagrazia Graziano, Marco Vacca, Alessandro Chiolerio and Maurizio Zamboni **"An NCL-HDL Snake-Clock-Based Magnetic QCA Architecture"**, IEEE transaction on nanotechnology, vol. 10, No. 5, september 2011, DOI 10.1109/TNANO.2011.2118229.

- Mariagrazia Graziano, Marco Vacca, Davide Blua and Maurizio Zamboni **"Asynchrony in Quantum-Dot Cellular Automata Nanocomputation: Elixir or Poison?"**, IEEE Design & Test of Computers, 2011, DOI 10.1109/MDT.2011.98.

- Marco Vacca, Mariagrazia Graziano and Maurizio Zamboni **"Asynchronous Solutions for Nano-Magnetic Logic Circuits"**, ACM Journal on Emerging Technologies in Computing Systems, Volume 7 Issue 4, December 2011, DOI 10.1145/2043643.2043645.

- Marco Vacca, Mariagrazia Graziano and Maurizio Zamboni **"NanoMagnetic Logic Microprocessor Hierarchical Power Model"**, IEEE transaction on vlsi circuits, in publishing, early access, DOI 10.1109/TVLSI.2012.2211903.

- Marco Vacca, Mariagrazia Graziano and Maurizio Zamboni **"Majority Voter Full Characterization for NanoMagnet Logic Circuits"**, IEEE transaction on nanotechnology, in publishing, early access, DOI 10.1109/TNANO.2012.2207965.

- Muhammad Awais, Marco Vacca, Mariagrazia Graziano and Guido Masera **"Quantum dot Cellular Automata Check Node Implementation for LDPC Decoders"**, IEEE transaction on nanotechnology, accepted publication.

## A.3 Books and books' chapters

- Mariagrazia Graziano, Marco Vacca and Maurizio Zamboni **"Magnetic QCA Design: Modeling, Simulation and Circuits"**, Cellular Automata Innovative Modelling For Science And Engineering, Intechweb.org, february 2011, ISBN 978-953-307-172-5,DOI 10.5772/15872.

# Appendix B

# How to write an article - Simple guidelines on how to write your first article

## B.1 Article General Organization

- **Title**
- **Authors list**
- **Abstract**
- **Keywords**
- **Introduction**
- **Basic concepts description**
- **Work description**
- **Conclusions and future works**
- **Acknowledgments**
- **Bibliography**

## B.2 Article Sections

### B.2.1 Title

- Generally it should be simple and short.

- Length: 1-2 lines (depending on the space available), but 1 line or less is better.
- The name must identify clearly the topic of the article.
- It should be easy to remember.
- In some case the name must be crazy enough to do a good impression on the reviewers.

### B.2.2 Authors List

- The name order is important (especially abroad).
- The first name is the name of who as done the majority of the work.
- If more authors have contributed equally to the work a * should be indicated near their names, and a note should be used to indicate this fact.
- The last author is normally the head of the group.
- If possible it is better to put not too many authors in journal publications.
- Number of authors is not relevant if the target of the article is a conference.

### B.2.3 Abstract

- It's probably the most important part.
- It should be short, simple and it must clearly explain the main concepts of the article.
- It should be composed by two short paragraph.
  - FIRST PARAGRAPH. It should give a general idea of the problem/technology that is analyzed in the article, highlighting its main advantages and what is already done in the literature, but at the same time saying what is the problem or the particular feature that is analyzed in the article, making a bridge with the second paragraph. Something like this for example: *"The pizza-generator is the most promising technology for the solution of humanity problems. It transforms Politicians into more useful Pizzas, that can be eaten by the hungry kids of Africa. However its high consumption of Politicians constitute a serious limitation to its use, since the number of Politicians in the world is limited."* It is ideally divided in two parts, the first one which identify the general topic of the article *"The*

*pizza-generator is the most promising technology for the solution of the problems of humanity. It transforms Politicians into more useful Pizzas, that can be eaten by the hungry kids of Africa.“*. The second part which describes the problem/part that will be analyzed in the article *“However its high consumption of Politicians constitute a serious limitation to its use, since the number of Politicians in the world is limited.“*

- SECOND PARAGRAPH. It uses the second part of the first paragraph as a bridge. It must describe the solution/proposal of the problem in the second part of the first paragraph and then finally it must say what it is done in the article, and its innovation, for example: *”A possible solution is the use of stock-exchange Speculators in place of Politicians, they are present in higher numbers in the world. In this work we propose an innovative pizza-generator that can works with many sources: Politicians, stoke-exchange Speculators, Bankers and generally with all the brainless, fat, old mummies that are destroying this world for their personal income. The revolutionary solution here proposed can lead the humanity into a brighter future, to the conquest of the rest of the universe.“* It is again divided in two parts. The first one which describes the solution to the problem described in the second part of the first paragraph *“A possible solution is the use of stock-exchange Speculators in place of Politicians, they are present in higher numbers in the world“*. The second part describing what is effectively done in the work **highlighting the innovation of the work** *“In this work we propose an innovative pizza-generator that can works with many sources: Politicians, stoke-exchange Speculators, Bankers and generally with all the brainless, fat, old mummies that are destroying this world for their personal incomes. The revolutionary solution here proposed can lead the humanity into a brighter future, to the conquest of the rest of the universe.“*

- ABSTRACT EXAMPLES

  - *The pizza-generator is the most promising technology for the solution of humanity problems. It transforms Politicians into more useful Pizzas, that can be eaten by the hungry kids of Africa. However its high consumption of Politicians constitute a serious limitation to its use, since the number of Politicians in the world is limited.*
    *A possible solution is the use of stock-exchange Speculators in place of Politicians, they are present in higher numbers in the world. In this work we propose an innovative pizza-generator that can works with many sources: Politicians, stoke-exchange Speculators, Bankers and generally with all the brainless, fat, old mummies that are destroying this world*

*for their personal income. The revolutionary solution here proposed can lead the humanity into a brighter future, to the conquest of the rest of the universe.*

- ...or if you want a more boring example... *In the years to come new solutions will be required to overcome the limitations of scaled CMOS technology. One approach is to adopt NanoMagnetic Logic Circuits, highly appealing for their extremely reduced power consumption. Despite the interesting nature of this approach, many problems arise when this technology is considered for real designs. The wire is the most critical of these problems from the circuit implementation point of view. It works as a pipelined interconnection, and its delay in terms of clock cycles depends on its length. Serious complications arise at the design phase, both in terms of synthesis and of physical design.*
  *One possible solution is the use of a delay insensitive asynchronous logic, Null Convention Logic (NCL). Nevertheless its use has many negative consequences in terms of area occupation and speed loss with respect to a Boolean version. In this article we analyze and compare different solutions: nanomagnetic circuits based on full NCL, mixed Boolean-NCL, and fully Boolean logic. We discuss the advantages of these logics, but also the issues they raise. In particular we analyze feedback signals, which, due to their intrinsic pipelined nature, cause errors that still have not found a solution in the literature. The innovative arrangement we propose solves most of the problems and thus soundly increases the knowledge of this technology. The analysis is performed using a VHDL behavioral model we developed and a microprocessor we designed based on this model, as a sound and realistic test bench.*

- **REMEMBER: the abstract is the most important part of the article, it can alone decide the acceptance/not acceptance of the article.**

- While the rest of the article should be written in an impersonal form, in some occasion it is good to use the "We" form. This can be useful at the end of the abstract or the introduction, where you can say "We have done this and this...".

### B.2.4 Keywords

- It contains a short list of words, related to the main ideas of the work, which will be used for indexing purposes.

- Use as example existing articles.

### B.2.5 Introduction

- It's the second most important part of the article...
- ...but its the most difficult to write.
- With the abstract it can decide the acceptance/not acceptance of the article (regardless of what you have written inside the article, regardless of the quality of the work).
- It can be seen as an extended version of the abstract.
- It can be ideally divided into three parts:
  - FIRST PART (corresponding to the first part of the first paragraph of the abstract). It contains a generic description of the technology/problem that is studied in the work. *"In the NanoMagnet based Logic (NML) digital values are represented using single domain nanomagnets (Fig. 1.A). If magnets are sufficiently small and are rectangularly shaped, they can assume only two stable magnetization states used to represent the logic values '0' and '1' [1]. Circuits are built placing magnets one near each other. Information propagates using the magnetic interaction among neighbor magnets. The basic logic gate is the Majority Voter (MV, Fig. 1.B), comprised of three input magnets surrounding a central element which performs the logic operation (see sec. II for background on NML). The value of the output magnet is equal to the value of the majority of the three inputs [1]. Although the maximum allowed frequency of this technology is low [2] (about 100 MHz if all constraints are taken into account, compared to THz for the molecular nearest counterpart [3]), NML is interesting because the expected power consumption is much lower than in CMOS circuits (about 100 times less) [4]. Moreover, due to their magnetic nature, they maintain the information stored also without power supply. Therefore this technology offers the possibility to combine logic and memory in the same device. As a consequence new way of developing logic circuits and their applications can be explored, with the possibility to further reduce power consumption."*
  - SECOND PART (corresponding more or less to the second part of the first paragraph of the abstract and to the first part of the second paragraph of the abstract). It contains the description of a particular problem/feature of the technology studied in the article and its possible solutions, both proposed in the literature and in the article itself. *"Many works in the literature analyze the behavior of the basic blocks of this technology*

*and, in some cases, how these blocks are influenced by magnets shapes and positions [1], [5], [6], [7]. However, no previous study considers with a thorough analysis the impact of the variation of some important parameters, as i) the distances among neighbor magnets, ii) the sizes of the magnets themselves, iii) the impact of these parameters on their switching time and energy consumption, and iv) the relations between the previously mentioned parameters and the clock physical organization. In this work, starting from our preliminary contribution in [8], we study the MV using low level micromagnetic simulators, OOMMF [9], and in particular NMAG [10], which allows not only a behavioral analysis, but also enables to extract quantitative data on timing and energy performance.“*

– THIRD PART (corresponding to the last part of the abstract). It contains the description of what is effectively done in the work, **highlighting the innovation over the existing literature**. It contains the motivation of the work, and it is the most important part of the introduction. It can contain a small summary of the article, with a brief description of the various sections (although it is not strictly required). The “We” form can be used in this part of the introduction. *“We simulate (section III) the gate in various conditions where we change distances among neighbor magnets, as well as their aspect ratio. The purpose of this analysis is to verify whether these circuits can be built using lithographic techniques that have a low resolution, are fast and allow for high volume production (i.e. Ultra Deep Ultraviolet Lithography). We then analyze how the MV behaves considering process variations (in section IV), because a good rejection process related errors highly increases the chances of using this technology. We also study (section V and VI) how the most important features of the gate, timing and energy dissipation, change due to variations in magnets sizes and distances. Finally, we discuss (in section VII) issues related to the fabrication of realistic gates considering the real structure of clock wires. We propose a modification of the clock wires and we achieve a solution that assures to obtain gates that correctly work without the need of complex magnets organization as previously proposed.“*

- Maximum 1-2 Figures can be inserted in this part to help to explain the basic concepts.

- Citations must be carefully inserted in this part. They must be equally distributed along the introduction covering all the parts, from the basics of the technology/problem described in the first part of the introduction, to the description of the problem/feature described in the article in the second part of the introduction.

- Citations must be inserted wherever a statement assumption is done without justifying it (i.e. *Although the maximum allowed frequency of this technology is low [2]...*).

- It is better to not insert too many references to the previous works done by yourself.

- The insertion of the citation is normally a very difficult task.

- Depending on the target of the article the introduction can be merged with the next section "Basic Concepts Description" to save space (typically in case of conferences or letters). In this case the first and second parts of the introduction must be enlarged with more details.

- The introduction can be schematically seen as something like "We are talking of... which is the most wonderful technology ever invented... which is normally studied in this way... however it suffer of these problems... and the previous works on the topic have these problems... as a consequence we propose this... which incredibly enhance the knowledge of the topic". It is a difficult task in which one must highlight the advantages of the technology studied and the previous work done, but highlighting at the same time the problems of the technology in the previous works, in order to create a link with the work done in this article highlighting its innovation with respect to the existing one.

- If you are not an expert it is better to write abstract, introduction and conclusions at last, while when you become expert probably you will start writing these parts at the beginning.

### B.2.6 Basic Concepts Description

- It contains the detailed description of the fundamental basics of the technology/problem studied in the work.

- It is an expansion of the first part of the introduction.

- All the relevant aspects of the technology/problem must be described. Relevant means all the aspects that are necessary to understand the work described in the article for a not-expert of the matter described in the article itself. This mean the who writes the article must start from the assumption that who is reading the article don't knows the technology/problem described. In other words this part of the article must be "Idiot-proof".

- It is normally convenient to describe with a good detail also part that are extensively described in previous works (of whom writes the article). Normally a citation is not enough, readers are lazy people that don't want to read thousands of articles to understand the one that they are reading. If it is mandatory to save space it is possible to say that a particular part of the basic theory is not covered for space reason, and that more detail can be found in the cited paper.

### B.2.7 Work Description

- It contains the core part of the work, where the problem is accurately described.
- Its length and its structure (number and organization of sections) must be decided according to the space available.
- However it is important to well specify everything, choosing a section structure that helps to easily understand the problem.
- The last part of each section should be contain link to the next section.
- This is normally the most easy part of the paper to write.
- It is important to use images to help to understand the description.
- The last section must contains the RESULTS obtained by the work.

### B.2.8 Conclusions and Future Work

- This part is quite difficult to write, but is normally easier than the abstract and the introduction.
- It can be divided in two parts.
  - FIRST PART. A short summary of the article, **highlighting the main achievement and innovation of the work** . *"Our contribution notably improves the practical knowledge on NML especially considering to the impact that technological implementation has on NML circuits. We showed that it is possible to obtain correctly working circuits if specific constraints are respected: i) with gaps of 40-50 nm between nanomagnets the logic gate considered behaves correctly, thus deep UV lithography becomes the preferred fabrication technique; ii) magnets aspect ratio not far from 2 is the best solution for behavior and performance; iii) small*

*magnets sizes assure a better rejection to process variations; iv) timing and energy consumptions are precisely related to magnets distances, sizes and input configurations, and values for a correct optimization are given; v) Majority Voter realistic implementation might require to satisfy considerably impractical constraints, that our proposed solution based on an alternative clock distribution technique can successfully overcome.“*

- SECOND PART. A brief description of the direction in which the research will continue in the future. *“We are working on the experimental validation of these results with focus on the analysis of the clock system, both from the simulation and the experimental point of view, as we believe the clock system being the real obstacle to a realistic implementation of this technology.”*

- In this section it is useful to use the “We” form.

### B.2.9 Acknowledgments

- Sometimes it is good or necessary to add an acknowledgement at the end of the article, before the bibliography.
- It consists of few lines where you thank a specific person or an institution.
- It is used mainly in two situations:
  - when the work is based on the work made by someone else that is not inserted in the author list at the beginning;
  - when the work is part of a founded project you must thank the institution that gives you the money.

### B.2.10 Bibliography

- It contains all the works cited in the article.
- Using latex it is important to use .bib files.
- Works must appear in the cited order.
- It is important to add enough citations in the article to cover all the important statements assumptions.
- The number of citations must be related to available space (normally at least 8-10 citations for a conference and 20-30 citations for a journal).

- Citations must be recent if possible.
- Also it is better to not add too many self-citations of previous works.

## B.3 Hints & Tips

### B.3.1 Article Structure

- The structure described above is only an example, it can be changed depending on the target (conference, journal, letter, book) and the space available.
- However it MUST contain at least:
  - Abstract
  - Introduction/State of the art
  - Work Description
  - Results/Conclusions
  - Bibliography
- Always look at other work published on the same journal/conference to find useful hints.
- Always download the template that every journal/conference has.

### B.3.2 Writing Order

- Write the different parts of the article in the order that you prefer, however here you find an advice on the step to follow:
  - Define the structure of the article according to the target, previous works on that target and the space available.
  - Choose what Figures to insert in the article and place them.
  - Writing the article is now a matter of describing the Figures inserted...
  - Write all the sections of the article:
    * if you are not-expert start with the description of the problem, write everything without bothering about space problems, then write the three troublesome parts (abstract, introduction and conclusions) and then refine the article and reduce the text to fit into the page limit.

    * If you are already an expert start writing the conclusions, then the abstract and finally the introduction and then the core part of the article, try to write everything keeping into account the page limits.

- A good idea is to stay one-page less than the page limit. After the first review you will have to make many modifications to the article, normally adding things (most of the time useless and with no correlation to the work itself...) to satisfy the needs of the reviewers.

### B.3.3 Language style

- Language can be a problem for not-native English speaker.

- Problems arises when who writes the article is a not-native English speaker and the reviewer is a native English speaker.

- Comments will say that your article is unreadable, that your English is terrible, that there are too many errors (and the funny thing is that most of the comments will have much more errors than your article :)...). It is probably true that you have to improve a little your writing style and your English, but DON'T WORRY you are not so bad as they say.

  – The fact is quite simple: if you are a not-native English speaker, you must speak English but probably you will never speak like an English (and where is the problem? **It is necessary to speak English, not to speak as an English**, if you are not English you don't have to kill your identity to satisfy the pride of the World-Dominators).

  – As a second things, we are scientist not book writers like Steven King so don't bother too much with writing style.

  – Finally a very important fact: with the diffusion of English all around the world, the language is changing, it is becoming International-English, the one that we speak that is quite different from the one spoken by native English speakers. So probably you are not wrong but it is the reviewer that is wrong. He must study the International English. It is easy to do nothing and criticize you while you have to spend an entire life learning a foreign and alien language...

- However here you can find few tips on the language style to use to avoid too many problems:

  – Use short and simple phrases.

- Repeat many time the same name instead of using pronouns or hidden subject.
- Don't use complex and elaborate phrases, speak like a 10 years old lad.

### B.3.4 Journals, Conferences, Letters, Book Chapters

- JOURNALS. Journals are the most important target for publications.
  - The page limit is normally quite high (from 8-10 pages to no limit at all).
  - Sometimes, like in magazine, the limit is not the number of pages itself but the number of words, figures and citations.
  - Since the space available is enough the full structure of the article here described can be followed.
  - When writing for a journal it is important to do some research on the previously published work. It is important to understand the type of works that are normally published, their style, their focus (experimental/simulative). This search gives to the writer useful hints on how to write the article or if it is/it is not the case to submit the work to this journal or it is better to change target.
  - Journals are classified using a value called "Impact Factor", that gives the overall importance of the journal. It is not always correlated to the quality of the works published, but, if possible, it is better to choose journals with an high impact factor. Clearly the quality of the work must match the impact factor, since to publish on journals with high impact factor is more difficult.
  - The publication process follow some steps:
    * paper submissions;
    * first answer (3-6 months);
    * then the paper can be accepted (rarely it is immediately accepted), it can be marked as "Major Revision", "Minor Revision" or it can be rejected;
    * "Major Revision" means that the article need major corrections to its structure;
    * "Minor Revision" means that the article requires only corrections to some details but not to the article structure;
    * "Rejected" means rejected... or better try with another journal, perhaps you will find better reviewers...

    - for each major or minor revision the article must be modified according to the reviewer comments and then resubmitted;
    - it is possible to receive many major/minor revisions in sequence (each takes more or less 2 months), so the acceptance time can be very long;
    - when the article is accepted it will be published on-line;
    - after the acceptance you will be required to make some stylistic changes to the article or to correct minor errors;
    - in this final submission it is possible to make some changes to the article without changing its core part, this changes includes adding/removing references or change the text/figure size/position to reduce space and to fit the page limits.
  - Journals can have an acceptance time quite long, from 6 months in the best case, to 1 year in the average case, to 2-3 years in the worst case.
  - Quite often journals have special numbers called "Special Issue". This are numbers dedicated to one specific topic. It is important to submit articles to these special issues because the publication time is normally much faster. Clearly it is necessary to have a work that fits the topic of the special issue...

- CONFERENCES. Conferences are useful to know others people working in your same sector.
  - The page limit is smaller, normally conference papers are made of 4-6 pages.
  - The review process is faster. It can be a simple accepted/rejected response. In other cases after receiving the first comments, if the paper is accepted small changes can be required.
  - One of the advantages of Conferences Papers is clearly their fast publication with respect to journals.
  - However conferences publications are normally evaluated much less than journals.

- LETTERS. They are particular types of journals, with faster publication times.
  - The page limit is normally lower than journals (from 2 to 5-6 pages).
  - Acceptance times are much faster (5-6 weeks).
  - The impact factor can be very high.

  - Not all the works are suited for a publication on a letter. It is fundamental a research on the previously published articles, to understand if the idea can be submitted to that journal, and how much high are the chances that the publication will be accepted.

- BOOKS. They are particular types of publications.
  - Books and books chapters have normally less limitations.
  - These kind of publications have normally another target than the other types of publications. They normally have a wider range, they cover not a specific problem but all the aspects of a technology/problem.

### B.3.5 Figures

- Figures are one of the most important part of an article. A good Figure says much more than any textual description.
- Generally speaking the Figure must be clear and easy to understand not only when it is seen on the PC display, but more importantly when the paper it is printed.
- Figures can be colored, but it is important that it can be understood also when seen in black&white mode.
- Lines of the drawing must be relatively large (no less than 2-3 point width).
- Fonts must be relatively big (no less than 20 if possible).
- It is important to not waste space inside the Figure, white space can be filled with bigger fonts to make Fonts more readable.
- Figures on two columns takes a lot of space, so try to not use them too much.
- Caption is very important in each Figure. It must be quite long and complete. **Figures must be perfectly understandable also without reading the article**.

### B.3.6 Latex or Word?

- LATEX
  - Easy to manage complex and articulated text.
  - It is not necessary to bother with bad text and figures alignment.

  - It requires a small experience.
  - Totally free.

- WORD
  - Are there any advantages to use this devil machine, a part from crashing your PC against the wall because Word keep placing your text and figures wherever it wants?
  - Are you a Fanboy Microsoft? Why do you need to pay to use something that works much worse than a free tool?
  - Use it only if the conference/journal target does not accept latex.

# Appendix C

# Program for NMAG automatic parametric analysis

## C.1 Main file

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#ifndef HEADERFILE_H
#define HEADERFILE_H
#include "headerfile.h"
#endif

int main(int argc, char *argv[]) {

  int i, j;
  char *temp;
  int width = 50;
  int height = 100;
  int thick = 20;
  int horizontal_dist = 40;
  int vertical_dist = 40;
  int minx = 10;
  int maxx = 90;
  int stepx = 10;
  int miny = 20;
  int maxy = 140;
  int stepy = 20;

  for (i=minx; i<=maxx; ) {
    for (j=miny; j<=maxy; ) {
      geometry(width,height,thick,horizontal_dist,vertical_dist,i,j);
      system("netgen␣-geofile=mv.geo␣-verycoarse
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣-meshfiletype=\"Neutral Format\"␣-meshfile=mv.neutral␣-batchmode");
      system("nmeshimport␣--netgen␣mv.neutral␣mv.nmesh.h5");
      system("nsim␣mv.py");
      /*if((temp = (char *)malloc(32)) != NULL)
         sprintf(temp, "%s%s%s%s%d%s%d", "mv␣*.txt", "␣", "mv", "_", i-10, "_", j-20);
      system(temp);
      free(temp);
```

```
      if((temp = (char *)malloc(32)) != NULL)
         sprintf(temp, "%s%s%s%s%d%s%d", "mv␣*.jpeg", "␣", "mv", "_", i-10, "_", j-20);
      system(temp);
      free(temp);*/
      // 2 magnet
      system("ncol␣mv␣time␣E_total_Py2␣E_ext_Py2␣>␣data_M21.txt");
      system("ncol␣mv␣time␣M_Py2_0␣␣M_Py2_1␣M_Py2_2␣>␣data_M22.txt");
      system("ncol␣mv␣time␣H_total_Py2_0␣H_total_Py2_1␣H_total_Py2_2␣>␣data_M23.txt");
      // 3 magnet
      system("ncol␣mv␣time␣E_total_Py3␣E_ext_Py3␣>␣data_M31.txt");
      system("ncol␣mv␣time␣M_Py3_0␣␣M_Py3_1␣M_Py3_2␣>␣data_M32.txt");
      system("ncol␣mv␣time␣H_total_Py3_0␣H_total_Py3_1␣H_total_Py3_2␣>␣data_M33.txt");
      // 4 magnet
      system("ncol␣mv␣time␣E_total_Py4␣E_ext_Py4␣>␣data_M41.txt");
      system("ncol␣mv␣time␣M_Py4_0␣␣M_Py4_1␣M_Py4_2␣>␣data_M42.txt");
      system("ncol␣mv␣time␣H_total_Py4_0␣H_total_Py4_1␣H_total_Py4_2␣>␣data_M43.txt");
      // 1 magnet
      system("ncol␣mv␣time␣E_total_Py1␣E_ext_Py1␣>␣data_M11.txt");
      system("ncol␣mv␣time␣M_Py1_0␣␣M_Py1_1␣M_Py1_2␣>␣data_M12.txt");
      system("ncol␣mv␣time␣H_total_Py1_0␣H_total_Py1_1␣H_total_Py1_2␣>␣data_M13.txt");
      // 5 magnet
      system("ncol␣mv␣time␣E_total_Py5␣E_ext_Py5␣>␣data_M51.txt");
      system("ncol␣mv␣time␣M_Py5_0␣␣M_Py5_1␣M_Py5_2␣>␣data_M52.txt");
      system("ncol␣mv␣time␣H_total_Py5_0␣H_total_Py5_1␣H_total_Py5_2␣>␣data_M53.txt");
      system("nsim␣graph.py");
      system("rm␣*_dat.h5");
      system("rm␣*_log.log");
      system("rm␣*_dat.ndt");
      if((temp = (char *)malloc(32)) != NULL)
         sprintf(temp, "%s%s%s%s%d%s%d", "mkdir", "␣", "mv", "_", i, "_", j);
      system(temp);
      free(temp);
      if((temp = (char *)malloc(32)) != NULL)
         sprintf(temp, "%s%s%s%s%d%s%d", "mv␣*.geo", "␣", "mv", "_", i, "_", j);
      system(temp);
      free(temp);
      sleep(5);
      if((temp = (char *)malloc(32)) != NULL)
         sprintf(temp, "%s%s%s%s%d%s%d", "mv␣*.txt", "␣", "mv", "_", i, "_", j);
      system(temp);
      free(temp);
      if((temp = (char *)malloc(32)) != NULL)
         sprintf(temp, "%s%s%s%s%d%s%d", "mv␣*.jpeg", "␣", "mv", "_", i, "_", j);
      system(temp);
      free(temp);
      j=j+stepy;
    }
   i=i+stepx;
  }
 return 0;
}
```

## C.2 Geometry creation file

```
#include <stdio.h>
#include <stdlib.h>
#ifndef HEADERFILE_H
#define HEADERFILE_H
#include "headerfile.h"
#endif
```

```
int geometry (int d, int h, int sp, int hd, int vd, int deltax, int deltay)
{
  FILE *fp;

  char name[]="mv.geo";
  int x1,y1,z1,x2,y2,z2,i;

  fp=fopen(name,"w");

  x1=0-(deltax/2);
  y1=0-(deltay/2);
  z1=0-(sp/2);
  x2=0+(deltax/2);
  y2=0+(deltay/2);
  z2=0+(sp/2);
  fprintf(fp,"algebraic3d\n\n");
  fprintf(fp,"# parallelepipeds consisting of 6 planes:\n\n");
  fprintf(fp,"solid par1 = plane (%d, %d, %d; 0, 0, -1)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; 0, -1, 0)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; -1, 0, 0)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; 0, 0, 1)\n",x2,y2,z2);
  fprintf(fp,"          and plane (%d, %d, %d; 0, 1, 0)\n",x2,y2,z2);
  fprintf(fp,"          and plane (%d, %d, %d; 1, 0, 0) -maxh=10.0;\n\n",x2,y2,z2);
  x1=0-(deltax/2);
  y1=h+vd-(deltay/2);
  z1=0-(sp/2);
  x2=0+(deltax/2);
  y2=h+vd+(deltay/2);
  z2=0+(sp/2);
  fprintf(fp,"solid par2 = plane (%d, %d, %d; 0, 0, -1)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; 0, -1, 0)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; -1, 0, 0)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; 0, 0, 1)\n",x2,y2,z2);
  fprintf(fp,"          and plane (%d, %d, %d; 0, 1, 0)\n",x2,y2,z2);
  fprintf(fp,"          and plane (%d, %d, %d; 1, 0, 0) -maxh=10.0;\n\n",x2,y2,z2);
  x1=-d-hd-(deltax/2);
  y1=0-(deltay/2);
  z1=0-(sp/2);
  x2=-d-hd+(deltax/2);
  y2=0+(deltay/2);
  z2=0+(sp/2);
  fprintf(fp,"solid par3 = plane (%d, %d, %d; 0, 0, -1)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; 0, -1, 0)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; -1, 0, 0)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; 0, 0, 1)\n",x2,y2,z2);
  fprintf(fp,"          and plane (%d, %d, %d; 0, 1, 0)\n",x2,y2,z2);
  fprintf(fp,"          and plane (%d, %d, %d; 1, 0, 0) -maxh=10.0;\n\n",x2,y2,z2);
  x1=0-(deltax/2);
  y1=-h-vd-(deltay/2);
  z1=0-(sp/2);
  x2=0+(deltax/2);
  y2=-h-vd+(deltay/2);
  z2=0+(sp/2);
  fprintf(fp,"solid par4 = plane (%d, %d, %d; 0, 0, -1)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; 0, -1, 0)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; -1, 0, 0)\n",x1,y1,z1);
  fprintf(fp,"          and plane (%d, %d, %d; 0, 0, 1)\n",x2,y2,z2);
  fprintf(fp,"          and plane (%d, %d, %d; 0, 1, 0)\n",x2,y2,z2);
  fprintf(fp,"          and plane (%d, %d, %d; 1, 0, 0) -maxh=10.0;\n\n",x2,y2,z2);
  x1=d+hd-(deltax/2);
  y1=0-(deltay/2);
```

```
  z1=0-(sp/2);
  x2=d+hd+(deltax/2);
  y2=0+(deltay/2);
  z2=0+(sp/2);
  fprintf(fp,"solid␣par5␣=␣plane␣(%d,␣%d,␣%d;␣0,␣0,␣-1)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣-1,␣0)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣-1,␣0,␣0)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣0,␣1)\n",x2,y2,z2);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣1,␣0)\n",x2,y2,z2);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣1,␣0,␣0)␣-maxh=10.0;\n\n",x2,y2,z2);
  x1=0-(d/2);
  y1=2*(h+vd)-(h/2);
  z1=0-(sp/2);
  x2=0+(d/2);
  y2=2*(h+vd)+(h/2);
  z2=0+(sp/2);
  fprintf(fp,"solid␣par6␣=␣plane␣(%d,␣%d,␣%d;␣0,␣0,␣-1)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣-1,␣0)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣-1,␣0,␣0)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣0,␣1)\n",x2,y2,z2);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣1,␣0)\n",x2,y2,z2);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣1,␣0,␣0)␣-maxh=10.0;\n\n",x2,y2,z2);
  x1=-2*(d+hd)-(d/2);
  y1=0-(h/2);
  z1=0-(sp/2);
  x2=-2*(d+hd)+(d/2);
  y2=0+(h/2);
  z2=0+(sp/2);
  fprintf(fp,"solid␣par7␣=␣plane␣(%d,␣%d,␣%d;␣0,␣0,␣-1)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣-1,␣0)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣-1,␣0,␣0)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣0,␣1)\n",x2,y2,z2);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣1,␣0)\n",x2,y2,z2);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣1,␣0,␣0)␣-maxh=10.0;\n\n",x2,y2,z2);
  x1=0-(d/2);
  y1=-2*(h+vd)-(h/2);
  z1=0-(sp/2);
  x2=0+(d/2);
  y2=-2*(h+vd)+(h/2);
  z2=0+(sp/2);
  fprintf(fp,"solid␣par8␣=␣plane␣(%d,␣%d,␣%d;␣0,␣0,␣-1)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣-1,␣0)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣-1,␣0,␣0)\n",x1,y1,z1);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣0,␣1)\n",x2,y2,z2);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣0,␣1,␣0)\n",x2,y2,z2);
  fprintf(fp,"␣␣␣␣␣␣␣␣␣␣and␣plane␣(%d,␣%d,␣%d;␣1,␣0,␣0)␣-maxh=10.0;\n\n",x2,y2,z2);
  for (i=0; i<=7; i++) {
    fprintf(fp,"tlo␣par%d;\n",i+1);
  }
  fclose(fp);
return 0;
}
```

# C.3 Simulation file

```
import nmag
import os
from nmag import SI, every, at

sim = nmag.Simulation()
```

```
# define magnetic material alloy of Permalloy
list=['Py1','Py2','Py3','Py4','Py5','Py6','Py7','Py8']
for i in range(0,8):
  j=i+1
  list[i] = nmag.MagMaterial(name="Py%d" %(j),
       Ms=SI(860e3, "A/m"),
       exchange_coupling=SI(13e-12, "J/m"),
       anisotropy=nmag.uniaxial_anisotropy(axis=[1,0,0], K1=SI(3.2e3, "J/m^3")))

# load mesh
sim.load_mesh("mv.nmesh.h5",
              [("par1", list[0]),("par2", list[1]),("par3", list[2]),
("par4", list[3]),("par5", list[4]),("par6", list[5]),("par7", list[6]),
("par8", list[7])],
              unit_length=SI(1e-9,"m")
              )


# Initial magnetization
sim.set_m([1,0,0])

# Input 001
def H_input(pos):
  x,y,z = pos
  newy=y*1e9
  newx=x*1e9
  # input 1
  if newx>=-25 and newx<=25 and newy>=230:
    return [0,-1e6, 0]
  # input 2
  elif newx>=-205 and newx<=-155 and newy>=-50 and newy<=50:
    return [0,1e6, 0]
  # input 3
  elif newx>=-25 and newx<=25 and newy<=-230:
    return [0,1e6, 0]
  else:
    return [1e6,0,0]

# External magnetic field region 2
def H_zone2(pos):
  x,y,z = pos
  newy=y*1e9
  newx=x*1e9
  # input 1
  if newx>=-25 and newx<=25 and newy>=230:
    return [0,-1e6, 0]
  # input 2
  elif newx>=-205 and newx<=-155 and newy>=-50 and newy<=50:
    return [0,1e6, 0]
  # input 3
  elif newx>=-25 and newx<=25 and newy<=-230:
    return [0,1e6, 0]
  elif newx>65 and newy>=-50 and newy<=50:
    return [1e6,0,0]
  else:
    return [0,0,0]

# External magnetic field region 3
def H_zone3(pos):
  x,y,z = pos
  newy=y*1e9
```

```
  newx=x*1e9
  # input 1
  if newx>=-25 and newx<=25 and newy>=230:
    return [0,-1e6, 0]
  # input 2
  elif newx>=-205 and newx<=-155 and newy>=-50 and newy<=50:
    return [0,1e6, 0]
  # input 3
  elif newx>=-25 and newx<=25 and newy<=-230:
    return [0,1e6, 0]
  else:
    return [0,0,0]

# Step of 10 nanoseconds
dt = SI(10e-12, "s") # corresponds to 10 nanoseconds

for i in range(0,80):
   sim.advance_time(dt*i)
   if i>=0 and i<10:
     sim.set_H_ext(H_input,SI ('A/m'))
   elif i>=10 and i<80:
     sim.set_H_ext(H_zone2,SI ('A/m'))
   else:
     sim.set_H_ext(H_zone3,SI ('A/m'))
   sim.save_data(fields='all')
```

## C.4 Graphs creation file

```
import nmag
import os
from nmag import SI, every, at

# 2 magnet
f=os.popen('gnuplot',"w")
print >>f, "set␣terminal␣jpeg"
# 1 graph
print >>f, "set␣output␣'E_tot_2.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Total␣Energy␣(Kg/m^2)'"
print >>f, "plot␣'data_M21.txt'␣using␣1:2␣with␣lines␣title␣'Total␣Energy␣Permalloy'"
# 2 graph
print >>f, "set␣output␣'E_ext_2.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'External␣Energy␣(Kg/m^2)'"
print >>f, "plot␣'data_M21.txt'␣using␣1:3␣with␣lines␣title␣'External␣Energy␣Permalloy'"
# 3 graph
print >>f, "set␣output␣'Magnetization_2.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Magnetization␣(A/m)'"
print >>f, "plot␣'data_M22.txt'␣using␣1:2␣with␣lines␣title␣'M_Py␣x',␣'data_M22.txt'
␣␣␣using␣1:3␣with␣lines␣title␣'M_Py␣y',␣'data_M22.txt'
␣␣␣using␣1:4␣with␣lines␣title␣'M_Py␣z'"
# 4 graph
print >>f, "set␣output␣'H_total_2.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Total␣Magnetic␣field␣(A/m)'"
print >>f, "plot␣'data_M23.txt'␣using␣1:2␣with␣lines␣title␣'H_total_Py␣x',␣'data_M23.txt'
␣␣␣using␣1:3␣with␣lines␣title␣'H_total_Py␣y',␣'data_M23.txt'
␣␣␣using␣1:4␣with␣lines␣title␣'H_total_Py␣z'"
```

```
# 3 magnet

# 1 graph
print >>f, "set␣output␣'E_tot_3.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Total␣Energy␣(Kg/m^2)'"
print >>f, "plot␣'data_M31.txt'␣using␣1:2␣with␣lines␣title␣'Total␣Energy␣Permalloy'"
# 2 graph
print >>f, "set␣output␣'E_ext_3.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'External␣Energy␣(Kg/m^2)'"
print >>f, "plot␣'data_M31.txt'␣using␣1:3␣with␣lines␣title␣'External␣Energy␣Permalloy'"
# 3 graph
print >>f, "set␣output␣'Magnetization_3.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Magnetization␣(A/m)'"
print >>f, "plot␣'data_M32.txt'␣using␣1:2␣with␣lines␣title␣'M_Py␣x',␣'data_M32.txt'
␣␣␣using␣1:3␣with␣lines␣title␣'M_Py␣y',␣'data_M32.txt'
␣␣␣using␣1:4␣with␣lines␣title␣'M_Py␣z'"
# 4 graph
print >>f, "set␣output␣'H_total_3.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Total␣Magnetic␣field␣(A/m)'"
print >>f, "plot␣'data_M33.txt'␣using␣1:2␣with␣lines␣title␣'H_total_Py␣x',␣'data_M33.txt'
␣␣␣using␣1:3␣with␣lines␣title␣'H_total_Py␣y',␣'data_M33.txt'
␣␣␣using␣1:4␣with␣lines␣title␣'H_total_Py␣z'"

# 4 magnet

# 1 graph
print >>f, "set␣output␣'E_tot_4.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Total␣Energy␣(Kg/m^2)'"
print >>f, "plot␣'data_M41.txt'␣using␣1:2␣with␣lines␣title␣'Total␣Energy␣Permalloy'"
# 2 graph
print >>f, "set␣output␣'E_ext_4.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'External␣Energy␣(Kg/m^2)'"
print >>f, "plot␣'data_M41.txt'␣using␣1:3␣with␣lines␣title␣'External␣Energy␣Permalloy'"
# 3 graph
print >>f, "set␣output␣'Magnetization_4.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Magnetization␣(A/m)'"
print >>f, "plot␣'data_M42.txt'␣using␣1:2␣with␣lines␣title␣'M_Py␣x',␣'data_M42.txt'
␣␣␣using␣1:3␣with␣lines␣title␣'M_Py␣y',␣'data_M42.txt'
␣␣␣using␣1:4␣with␣lines␣title␣'M_Py␣z'"
# 4 graph
print >>f, "set␣output␣'H_total_4.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Total␣Magnetic␣field␣(A/m)'"
print >>f, "plot␣'data_M43.txt'␣using␣1:2␣with␣lines␣title␣'H_total_Py␣x',␣'data_M43.txt'
␣␣␣using␣1:3␣with␣lines␣title␣'H_total_Py␣y',␣'data_M43.txt'
␣␣␣using␣1:4␣with␣lines␣title␣'H_total_Py␣z'"

# 1 magnet

# 1 graph
print >>f, "set␣output␣'E_tot_1.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Total␣Energy␣(Kg/m^2)'"
print >>f, "plot␣'data_M11.txt'␣using␣1:2␣with␣lines␣title␣'Total␣Energy␣Permalloy'"
# 2 graph
```

```
print >>f, "set␣output␣'E_ext_1.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'External␣Energy␣(Kg/m^2)'"
print >>f, "plot␣'data_M11.txt'␣using␣1:3␣with␣lines␣title␣'External␣Energy␣Permalloy'"
# 3 graph
print >>f, "set␣output␣'Magnetization_1.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Magnetization␣(A/m)'"
print >>f, "plot␣'data_M12.txt'␣using␣1:2␣with␣lines␣title␣'M_Py␣x',␣'data_M12.txt'
␣␣␣using␣1:3␣with␣lines␣title␣'M_Py␣y',␣'data_M12.txt'
␣␣␣using␣1:4␣with␣lines␣title␣'M_Py␣z'"
# 4 graph
print >>f, "set␣output␣'H_total_1.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Total␣Magnetic␣field␣(A/m)'"
print >>f, "plot␣'data_M13.txt'␣using␣1:2␣with␣lines␣title␣'H_total_Py␣x',␣'data_M13.txt'
␣␣␣using␣1:3␣with␣lines␣title␣'H_total_Py␣y',␣'data_M13.txt'
␣␣␣using␣1:4␣with␣lines␣title␣'H_total_Py␣z'"

# 5 magnet

# 1 graph
print >>f, "set␣output␣'E_tot_5.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Total␣Energy␣(Kg/m^2)'"
print >>f, "plot␣'data_M51.txt'␣using␣1:2␣with␣lines␣title␣'Total␣Energy␣Permalloy'"
# 2 graph
print >>f, "set␣output␣'E_ext_5.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'External␣Energy␣(Kg/m^2)'"
print >>f, "plot␣'data_M51.txt'␣using␣1:3␣with␣lines␣title␣'External␣Energy␣Permalloy'"
# 4 graph
print >>f, "set␣output␣'Magnetization_5.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Magnetization␣(A/m)'"
print >>f, "plot␣'data_M52.txt'␣using␣1:2␣with␣lines␣title␣'M_Py␣x',␣'data_M52.txt'
␣␣␣using␣1:3␣with␣lines␣title␣'M_Py␣y',␣'data_M52.txt'
␣␣␣using␣1:4␣with␣lines␣title␣'M_Py␣z'"
# 4 graph
print >>f, "set␣output␣'H_total_5.jpeg'"
print >>f, "set␣xlabel␣'time␣(seconds)'"
print >>f, "set␣ylabel␣'Total␣Magnetic␣field␣(A/m)'"
print >>f, "plot␣'data_M53.txt'␣using␣1:2␣with␣lines␣title␣'H_total_Py␣x',␣'data_M53.txt'
␣␣␣using␣1:3␣with␣lines␣title␣'H_total_Py␣y',␣'data_M53.txt'
␣␣␣using␣1:4␣with␣lines␣title␣'H_total_Py␣z'"
f.flush()
```

## C.5 Header file

```
// Declearation functions
int geometry (int d, int h, int sp, int hd, int vd, int deltax, int deltay);
```

## C.6 Make file

```
CC = gcc
REQ = geo.c main.c
OBJ = geo.o main.o
```

```
simu: $(REQ)
        @$(CC) -c -o main.o main.c -lm
        @$(CC) -c -o geo.o geo.c
        @$(CC) -o run $(OBJ) -lm
        @echo "Compiled. Type ./run to run"

debug: $(REQ)
        @$(CC) -g -c -o main.o main.c -lm
        @$(CC) -g -c -o geo.o geo.c
        @$(CC) -g -o run $(OBJ) -lm
        @echo "(Debug) Compiled. Type ./run to run"

clean:
        @rm -f *.o
        @rm -f run
        @rm -f *.swp
        @rm -f *~
```

# Bibliography


[1] L. Di Crescenzo. Design And Implementation of an Automatic Layout Generator for NanoMagnet Technology Circuits . Master's thesis, November 2012.

[2] M. Niemier and al. Nanomagnet logic: progress toward system-level integration. *J. Phys.: Condens. Matter*, 23:34, November 2011.

[3] D.B. Carlton, N.C. Emley, E. Tuchfeld, and J. Bokor. Simulation Studies of Nanomagnet-Based Logic Architecture. *Nanoletters*, 8(12):4173–4178, November 2008.

[4] E. Varga, G. Csaba, G.H. Bernstein, and W. Porod. Implementation of a Nanomagnetic Full Adder Circuit. *2011 11th IEEE International Conference on Nanotechnology*, August 2011.

[5] M. Graziano, M. Vacca, A. Chiolerio, and M. Zamboni. A NCL-HDL Snake-Clock Based Magnetic QCA Architecture. *IEEE Transaction on Nanotechnology*, (10):DOI:10.1109/TNANO.2011.2118229.

[6] M.T. Niemier, E. Varga, G.H. Bernstein, W. Porod, M.T. Alam, A. Dingler, A. Orlov, and X.S. Hu. Shape Engineering for Controlled Switching With Nanomagnet Logic. *IEEE Transactions on Nanotechnology*, 11(2):220–230, March 2012.

[7] J. Wang. Emerging Technologies For Biosequence Analysis . Master's thesis, November 2012.

[8] J.L. Schiff. *Cellular Automata: A Discrete View of the World*. Wiley & Sons, 2007.

[9] C.S. Lent, P.D. Tougaw, W. Porod, and G.H. Bernstein. Quantum cellular automata. *Nanotechnology*, 4:49–57, 1993.

[10] P.D. Tougaw and C.S. Lent. Dynamic behavior of quantum cellular automata. *Journal Of Applied Physics*, (80):4722–4736, 1996.

[11] R.K. Kummamuru, A.O. Orlov, R. Ramasubramaniam, C.S. Lent, G.H. Bernstein, and G.L. Snider. Operation of a Quantum-dot Cellular Automata (QCA) shift register and analysis of errors. *IEEE Trans. On Electron Devices*, 50:1906, 2003.

[12] A.I. Csurgay, W. Porod, and C.S. Lent. Signal processing with near-neighborcoupled time-varying quantum-dot arrays. *IEEE Transaction On Circuits and Systems*, 47(8):1212–1223, 2000.

[13] A.O. Orlov, R.K. Kummamuru, R. Ramasubramaniam, C.S. Lent, G.H. Bernstein, and G.L. Snider. Clocked quantum-dot cellular automata devices: experimental studies. *Proceedings of the 2001 1st IEEE Conference on Nanotechnology, 2001. IEEE-NANO 2001*, pages 425–430, 2001.

[14] M.T. Niemier, M.J. Kontz, and P.M. Kogge. A Design of and Design Tools for a Novel Quantum Dot Based Microprocessor. *Proceedings of the 37th Annual Design Automation Conference*, 2000.

[15] A. Khitun and K.L. Wang. Multi-functional edge driven nano-scale cellular automata based on semiconductor tunneling nano-structure with a self-assembled quantum dot layer. *Superlattices and Microstructures*, 37(1):55–76, January 2005.

[16] C.G. Smith, S. Gardelis, A.W. Rushforth, R. Crook, J. Cooper, D.A. Ritchie, E.H. Linfield, Y. Jin, and M. Pepper. Realization of quantum-dot cellular automata using semiconductor quantum dots. *Superlattices and Microstructures*, 34(3-6):195–203, 2003.

[17] C.S. Lent and B. Isaksen. Clocked Molecular Quantum-Dot Cellular Automata. *IEEE Transactions on Electron Devices*, 50(9):1890–1896, September 2003.

[18] U. Lu and C.S. Lent. Theoretical Study of Molecular Quantum-Dot Cellular Automata. *Journal of Computational Electronics - Springer*, 4:115–118, 2005.

[19] H. Qi, S. Sharma, Z. Li, G.L. Snider, A.O. Orlov, C. S. Lent, and T.P. Fehlner. Molecular Quantum Cellular Automata Cells. Electric Field Driven Switching of a Silicon Surface Bound Array of Vertically Oriented Two-Dot Molecular Quantum Cellular Automata. *Journal Of The American Chemical Society*, 125(49):15250–15259, 2003.

[20] J. Jiao, G.J. Long, F. Grandjean, A.M. Beatty, and T.P. Fehlner. Building blocks for the molecular expression of quantum cellular automata. Isolation and characterization of a covalently bonded square array of two ferrocenium and two ferrocene complexes. *Journal of the American Chemical Society*, 125(25):7522–7523, 2003.

[21] G. Csaba and W. Porod. Simulation of Filed Coupled Computing Architectures based on Magnetic Dot Arrays. *J. of Comp. El., Kluwer,*, 1:87–91, 2002.

[22] M.T. Alam, M.J. Siddiq, G.H. Bernstein, M.T. Niemier, W. Porod, and X.S. Hu. On-chip Clocking for Nanomagnet Logic Devices. *IEEE Transaction on Nanotechnology*, 2009.

[23] J. Das, S.M. Alam, and S. Bhanja. Low Power Magnetic Quantum Cellular Automata Realization Using Magnetic Multi-Layer Structures. *J. on Emerging and Selected Topics in Circuits and Systems*, 1(3), September 267-276.

[24] M. S. Fashami, J. Atulasimha, and S. Bandyopadhyay. Magnetization Dynamics, Throughput and Energy Dissipation in a Universal Multiferroic Nanomagnetic Logic Gate with Fan-in and Fan-out. *Nanotechnology*, 23(10), February 2012.

[25] N. Rizos, M. Omar, P. Lugli, G. Csaba, M. Becherer, and D. Schmitt-Landsiedel. Clocking Schemes for Field Coupled Devices from Magnetic Multilayers. In *International Workshop on Computational Electronics*, pages 1–4, Beijin, China, 2009. IEEE.

[26] M.T. Alam, J.DeAngelis, M. Putney, X.S. Hu, W. Porod, M. Niemier, and G.H. Bernstein. Clock Scheme for Nanomagnet QCA. In *International Conference on Nanotechnology*, pages 403–408, Hong Kong, 2007. IEEE.

[27] Marco Vacca. Nanoarchitectures based on magnetic QCA. Master's thesis, Politecnico di Torino, 2008.

[28] M. Graziano, A. Chiolerio, and M. Zamboni. A Technology Aware Magnetic QCA NCL-HDL Architecture. In *International Conference on Nanotechnology*, pages 763–766, Genova, Italy, 2009. IEEE.

[29] M. Mascarino. Analysis and simulation of circuits based magnetic QCA. Master's thesis, Politecnico di Torino, November 2009.

[30] K.M. Fant and S.A. Brandt. NULL Convention Logic$^{TM}$, A Complete and Consistent Logic for Asynchronous Digital Circuit Synthesis. In *International Conference on Application Specific Systems*, pages 261–273, Chicago-Illinois, USA, 1996. IEEE.

[31] E. Tabrizizadeh, H.R. Mohaqeq, and A. Vafaei. Designing QCA Delay-Insensitive Serial Adder. *Proc. IEEE Int. Conf. on emerging trends in Engineering and Technology*, 2008.

[32] Marco Ottavi and al. HDLQ: A HDL Environment for QCA Design. *ACM J. on Emerging Tech. in Comp. Systems*, 2(4):243–261, 2006.

[33] E.W.Johnson J.R.Janulis S. Henderson and P.D. Tourgaw. Incorporating Standard CMOS Design Process Methodologies into the QCA Logic Design Process. *IEEE Transaction on Nanotechnology*, 3(1):2–9, 2004.

[34] W. Porod. Magnetic Logic Devices Based on Field-Coupled Nanomagnets. *Nano & Giga*, 2007.

[35] C. Augustine, X. Fong, B. Behin-Aein, and K. Roy. Ultra-Low Power Nano-Magnet Based Computing: A System-Level Perspective. *IEEE Transaction on Nanotechnology*, 10(4):778–788, 2011.

[36] K. Walus, M. Mazur, G. Schulhof, and G.A. Jullien. Simple 4-Bit Processor Based On Quantum-Dot Cellular Automata (QCA). *Intl. Conf. on Application-Specific Systems, Architecture and Processors*, 2005.

[37] J. Huang and F. Lombardi. *Design and Test of Digital Circuits by Quantum-Dot Cellular Automata*. Artech House Publishers, Boston/London, 2007.

[38] Mentor Graphics. http://www.modelsim.com.

[39] G. Csaba and W. Porod. Behavior of Nanomagnet Logic in the Presence of Thermal Noise. In *International Workshop on Computational Electronics*, pages 1–4, Pisa, Italy, 2010. IEEE.

[40] J. Das, S.M. Alam, and S. Bhanja. Ultra-Low Power Hybrid CMOS-Magnetic Logic Architecture . *Trans. on Computer And Systems*, 2011.

[41] D.K. Karunaratne and S. Bhanja. Study of single layer and multilayer nanomagnetic logic architectures . *Journal Of Applied Physics*, (111), 2012.

[42] M. Vacca and al. Asynchronous Solutions for Nano-Magnetic Logic Circuits. *ACM J. on Emerging Tech. in Comp. Systems*, 7(4), December 2011.

[43] M. Crocker, X.S. Hu, and M.T. Niemier. Design and Comparison of NML Systolic Architectures . *Nanoarch*, 2010.

[44] G. Causapruno. Analysis and Optimization of Parallel Processing Architectures for Nanotechnologies . Master's thesis, November 2012.

[45] M.J. Donahue and D.G. Porter. OOMMF User's Guide, Version 1.0. Technical Report Interagency Report NISTIR 6376, National Institute of Standards and Technology, Gaithersburg, September 1999.

[46] T. Fischbacher, M. Franchin, G. Bordignon, and H. Fangohr. A Systematic Approach to Multiphysics Extensions of Finite-Element-Based Micromagnetic Simulations: Nmag. *IEEE Transactions on Magnetics*, 43(6):Available on–line, 2007.

[47] T. Teodosio and L. Sousa. QCA-LG: A tool for the automatic layout generation of QCA combinational circuits. *Norchip conference*, 2007.

[48] K. Walus, T.J. Dysart, G.A. Jullien, and R.A. Budiman. QCADesigner: A Rapid Design and Simulation Tool for Quantum-Dot Cellular Automata. *IEEE Transaction on Nanotechnology*, 3(1), March 2004.

[49] S. Frache, M. Graziano, and M. Zamboni. A Flexible Simulation Methodology and Tool for Nanoarray-based Architectures. *IEEE International Conference on Computer Design*, pages 60–67, October 2010.

[50] S. Frache and al. ToPoliNano: Nanoarchitectures Design Made Real. *IEEE NANOARCH*, 2012.

[51] E.H. Sentovich, K.J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P.R. Stephan, R.K. Brayton, and A.L. Sangiovanni-Vincentelli. SIS: A System for Sequential Circuit Synthesis. Technical report, EECS Department, University of California, Berkeley, 1992.

[52] R. Ravichandran and al. Partitioning and placement for buildable QCA circuits . *DAC*, 1, 2005.

[53] W.J. Chung and al. Node duplication and routing algorithms for quantum-dot cellular automata circuits . *IEE Proc. on Circ., Dev. and Sys.*, 153(5), 2006.

[54] C. Sechen and A. Sangiovanni-Vincentelli. The TimberWolf placement and routing package. *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, 20(2):510–522, April 1985.

[55] D. Wang. Novel Routing Schemes for IC Layout Part I: Two-Layer Channel Routing . *Design Automation Conference*, 1991.

[56] Alexandra Imre. *Experimental study of nanomagnets for Quantum-dot cellular automata(MQCA)logic applications*. PhD thesis, University of Notre Dame, Notre Dame, Indiana, December 2005.

[57] D. Bisero, P. Cremon, M. Madami, S. Tacchi, G. Gubbiotti, G. Carlotti, and A.O. Adeyeye. Nucleation and Propagation of Vortex States in Dense Chains of Regular Particles. *Magnet2011, 2nd Italian conference on magnetism*, february 2011.

[58] Comsol Multiphysics. http://www.comsol.com/.

[59] J. Atulasimha and S. Bandyopadhyay. Hybrid spintronic/straintronics: A super energy efficient computing scheme based on interacting multiferroic nanomagnets . *2012 12th IEEE International Conference on Nanotechnology*, August 2012.

[60] G. Csaba, P. Lugli, and W. Porod. Power Dissipation in Nanomagnetic Logic Devices. In *International Conference on Nanotechnology*, pages 346–348, Munic, Germany, 2004. IEEE.

[61] M. Vacca and al. Majority Voter Full Characterization for Nanomagnet Logic Circuits. *IEEE T. on Nanotechnology*, 11(5), September 2012.

[62] T. Chung, S. Keller, and G.A. Carman. Electric-field-induced reversible magnetic single-domain evolution in a magnetoelectric thin film . *Applied Physics Letter*, (94), 2009.

[63] K. Roy, S. Bandyopadhyay, and J. Atulasimha. Switching dynamics of a magnetostrictive single- domain nanomagnet subjected to stress . *Phys. Rev. B*, (83):1–15, 2011.

[64] S. Guillon, D. Saya, L. Mazenq, L. Nicu, C. Soyer, J. Costecalde, and D. Remiens. Lead-zirconate titanate (PZT) nanoscale patterning by ultraviolet-based lithography lift-off technique for nano-electromechanical systems applications . *2011 International Symposium on Piezoresponse Force Microscopy and Nanoscale Phenomena in Polar Materials*, July 2011.

[65] C. Huang, Y. Chen, Y. Liang, T. Wu, H. Chen, and W. Chao. Fabrication of Nanoscale PtOx /PZT/PtOx Capacitors by E-beam Lithography and Plasma Etching with Photoresist Mask . *Electrochemical and Solid-State Letters*, 2006.

[66] W. Scholz, J. Fidler, T. Schrefl, D. Suess, R. Dittrich, H. Forster, and V. Tsiantos. Scalable Parallel Micromagnetic Solvers for Magnetic Nanostructures. *Comp. Mat. Sci.*, (28):366–383, 2003.

[67] International Technology Roadmap of Semiconductors, 2010. http://public.itrs.net.

[68] A. Chiolerio, M. Quaglio, A. Lamberti, F. Celegato, D. Balma, and P. Allia. Magnetoelastic coupling in multilayered ferroelectric/ferromagnetic thin films: A quantitative evaluation . *Applied Surface Science*, 258:8072–8077, 2012.

[69] A. Magni, F. Celegato, M. Coisson, E.S. Olivetti, M. Pasquale, and C.P. Sasso. Magnetization Properties of FeTb Thin Films . *IEEE Trans. on Magnetics*, 46(2), February 2010.