Politecnico di Torino

Dipartimento di Elettronica e Telecomunicazioni

Corso di Laurea Magistrale in Ingegneria Elettronica

Master Thesis

Reconfigurable standard-cell concept in Molecular Field-Coupled Nanocomputing

Supervisors:

Prof. Mariagrazia Graziano

Prof. Gianluca Piccinini

Candidate:

Christian Fabiano

December 2020

"Perché hai smesso di giocare?"
"Perché sono cresciuto"
"Beh, anche i bambini giocano a fare gli adulti"

Acknowledgements

For many of us, students, the greeting of our academic career is happening behind a computer screen, but at least my grandmother can easily assist the graduation ceremony. Five years ago, nobody could have imagined this strange ending of the university experience.

Even if unusual, I am excited about how things went and of the people I met. I would like to thank all the Polito professors for their strong passion, and a special thank goes to Prof. Graziano and Prof. Piccinini, and not only for what concerns electronics. I would like to thank them for the life experience they taught me, which I couldn't have found in books. A personal thank goes also to Yuri, that has wisely guided me during this work.

During my university career, I had also the chance to read the acknowledgements of my friends that graduated before me. I couldn't wait for the day in which my turn would have come: during these five years, I thought a lot about the friends I wanted to thank. Given the distance, I am not sure if those people will ever read this page, this is why I have decided to not write here their names. I think it would be nice if I could thank them eye to eye, one day.

There is, however, an exception. I feel it is the right moment to thank a person that will not be able to read this page and, unfortunately, I will not be able to thank in the first person, either, due to his early death from cancer a year ago.
Giovanni Pacifico has been my professor of Electronics during high school. His passion and dedication to the electronic disciplines have been a huge encouragement for me, and have addressed, among the other things, even what I have been studying for the last five years.

Thanks again, professor.

Abstract

Year after year, the limits of MOS technologies are becoming more and more tightening [1]. Due to the nanoscale effects of transistors and interconnections, new designs require a lot of effort to keep the performance-increasing trend: thermal and electrical uncertainties have to be counteracted at high design-abstraction levels. Moreover, only a few factories in the world are able to manufacture the latest-technology chips. This is why a key factor of a new technology could be, among the other factors, an easier and faster manufacturing. This could also promote a manufacturing cost reduction, allowing more factories to build state-of-the-art chips.
In this framework, molecular electronics plays a central role. Molecular Field-Coupled Nanocomputing (FCN) is a novel computing paradigm that implements, through molecules, the Quantum-Dot Cellular Automata (QCA) paradigm [2]. Molecular FCN binds a logical state to the charge distributions of molecules (Bisferrocene) and has demonstrated correct logic computation and reliable information propagation [3].
In order to further improve the ease of manufacturing, a puzzle-inspired 2D review of Molecular FCN is introduced. Instead of placing single molecules along a precise pattern, which requires a very high technological resolution, this new paradigm requires only a simple and squared self-assembled monolayer (SAM), which results in a grid of molecules placed on a gold substrate. The gold substrate serves also as an electrode plate: vertical electric fields are originated between the substrate and small square electrodes that lie above the molecules. The groups of molecules between these electrodes can hold a logical state. By turning on and off electrodes in adjacent areas, logic operations can be performed. Different models and electrodes arrangements are therefore discussed. The aim of the study was to define a molecular standard-cell concept that takes care of mapping the logic gates (NAND, majority voters, inverters) to the molecular structure. Moreover, electrical characteristics such as area and power dissipation have been compared with state-of-art FinFET technology: the new approach brings a reduction of area and power dissipation up to more than 50%. This review also highlights the new possibilities of reconfigurability of the logic operations (similar to FPGA), which could happen even at run-time.

List of Tables

List of Figures

Contents

Chapter 1

Computing state-of-the-art

When thinking of digital electronics and the technology behind it, it is a common standpoint to note how that starte-of-the-art technology is being pushed to its limits: this chapter shortly recalls the reasons behind new technologies' research and highlights a few of current research topics.

1.1 Moore's law

Moore's law predicts the doubling of transistors inside an integrated circuit every two years. It has been until today a reliable way to forecast the evolution of chips [1].

1.1.1 The golden age

Since Moore's law has been enunciated (1965), industries were able to comply with it without difficulty [1]. The transistors, building blocks of digital electronics, could easily be miniaturized: it was enough to adjust the levels of doping accordingly. Back in the years 1970-2000 the area of the chip was almost the only factor to influence its cost. The decrease of transistor dimensions allowed either to reduce the costs (at the same performance) or to increase the number of transistors and performance (at the same cost). Moreover, scaling policies had as a goal the increase of the current flowing into transistor's channels, which means faster loading of parasitic capacitance, followed by an increase of the frequency of operation.

1.1.2 Today's perspective

Today, the highest concern of chips is not anymore the area, but the power dissipation [7]. To avoid burn-out of the die, different areas of the design have to be selectively turned off. It is appropriate to say that *all the chickens are coming home to roost*: there is several of disciplines in which the limits of current technology are appearing. These include manufacturing difficulties and non-ideality of operations. The most important MOS non-ideality can be summarized as the following:

- Short-Channel Effects (SCE)
- Drain-Induced Barrier Lowering (DIBL)
- Gate tunneling (leakage)
- Technological Random Doping Fluctuations (RDF)

Short-Channel Effects (SCE)

Short-Channel Effects is the name given to a set of non-ideality that arise from reducing the gate length. The main feature is the so-called *threshold voltages roll-off*: the on-turning point of transistors is displaced from the expected value, and its value inherits the uncertainty on the gate length due to manufacturing. This circumstance happens because most of electrons charge in the channel is not anymore driven by the gate but also by source and drain areas.

Drain-Induced Barrier Lowering (DIBL)

Drain-induced barrier represents another way to look at the same aspect described by SCE. Since the gate is not long enough, source and drain take control over the channel's charge and leave more room for drain-source current leakage.

Gate tunneling

Gate tunneling represents an additional source of leakage in a MOS transistor that is manifested when the thickness of the oxide (between the metal and the silicon) goes under a certain value. The reduction in oxide thickness can help the control of the gate on the channel, but leaves room for tunneling of electrons between the metal and the substrate.

Random Doping Fluctuations (RDF)

When the size of source and drain regions reach the nanoscale dimension, it is easier to have a non uniform doping between different transistors. Since the number of silicon atoms in the channel is very low, it is enough a single more atom of dopant to change the transistor's characteristics. This uncertainty coming from manufacturing goes under the name of *Random Doping Fluctuations*.

Mock scaling of transistors

The reduction of transistors size cannot, however, go over a certain amount, since it meets the quantum atomic properties. For a couple of years, the name given to technology node has lost its link with real technology parameters, making every new technology node being called simply "0.7x of whatever it was before" (IRDS, Executive Summary, 2020).

1.2 Beyond CMOS

To overcome the scaling problems of the current silicon technologies, two main approaches have been defined:

- More than Moore
- Beyond CMOS

The first approach (More than Moore) relies still on CMOS and its evlutions (FinFET, GAAFET), but extends its sight towards solutions such as *heterogeneous integration* and *3D integration*. More than Moore is with a high change the approach that is going to be followed for the next generation chips, which makes of it the short-term solution. What is treated in this work is instead what could be a longer-term solution: *Beyond CMOS*. This last category includes all new devices and technologies that exploit totally different physical observable than the classic *current-based* transistors, and they include:

- Quantum-Dot Cellular Automata (QCA)
- Field-Coupled Nanocomputing (FCN)
- Tunneling Field-Effect Transistor (TFET)
- Single-Electron Transistor (SET)

The non-charge based category, instead, include devices that do not exploit the charge as physical observable, but other quantities such as photons or magnetic fields:

- Spintronics
- Photonics
- Etc...

According to the Beyond CMOS approach, if a new device is to be built from scratch using differet physical quantities, it will probably not be able to compete with CMOS: it should instead come together with a new architecture. It is believed that research should also focus on new architectural applications that could make better use of the new devices.

Chapter 2

Field Coupled Nanocomputing

Among the charge-based devices that belong to Beyond CMOS, an interesting computing paradigm is represented by Field-Coupled Nanocomputing (FCN). It includes all the devices that are able to interact with nearby devices by means of electric or magnetic fields. Among the FCN paradigms, *Quantum-Dot Cellular Automata (QCA)* is one of the most interesting candidates [8], and is the starting point of this work. According to QCA, the information is propagated through building blocks arranged in array form [4]. However, since this research overcomes some of the QCA concepts, all references to QCA will simply be indicated as FCN.

2.1 Basic concepts

The concept behind Field Coupled Nanocomputing exploits the possibility to perform computing using electric charges/fields instead of currents. The FCN device is a structured charge container. The classic structure is represented by a square with four "holes" called *dots*. A container is shown in fig. 2.1 [8] [9].

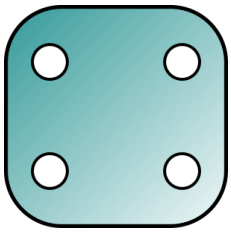


Figure 2.1: A basic FCN (QCA) container

2.1.1 Logic states

A *dot* is a vessel able to host a charge (or other physical observables). Depending on the presence of charge inside the vessel, the dot is either empty (represented with an empty circle)

or filled (represented by a colored circle). The information that is represented by the container relies on where the filled dots are localized [8]. A non-zero tunneling probability between quantum dots provides the switching capability [10]. Assuming that only two charges are placed inside the container, they will arrange on a diagonal, according to Coulomb repulsion. In fact, in order to have an equilibrium state for the information, only two out of the four dots are occupied by the charges. As an example, the logic value "0" can be associated to a container described by [charged empty; empty charged] and the logic value "1" can be associated to a container described by [empty charged; charged empty]. The association is shown in fig. 2.2.

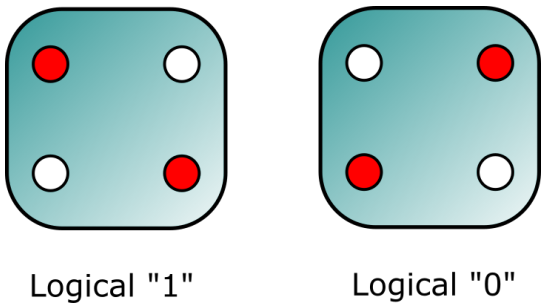


Figure 2.2: FCN diagonal states

2.1.2 Coupling

If placed one next to each other, two containers will experience an interaction. The charges inside the first container will generate an electrostatic field, that will couple with the field exercised by the charges in the second container. Assuming that one of the two has a "fixed" configuration, it will apply a force and possibly change the second container's configuration. Due to this force, and its ability to affect the a containers' associated information, it is possible to allow for information propagation and to perform logic operations.

2.2 Logic propagation

Since the localized charges produce a field, the nearby containers will react to the other's diagonal configuration. The arising coupling allows for logic operations and logic information propagation. Assuming that charges in first container are fixed, and that charges in the containers on the right are floating, those will adapt to a minimum energy configuration. As a chain effect, the containers will copy the previous containers and propagate the information. Fig 2.3 illustrates this concept.

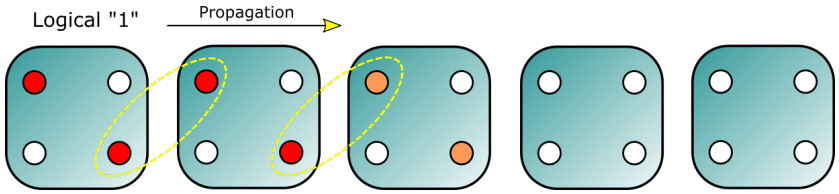


Figure 2.3: Coupling in a logic wire

2.3 Logic operations

Exploiting the coupling, logic gates such as *inverter* and *majority voter* are being implemented in FCN through particular geometric arrangements.

2.3.1 Inverter

In logic theory, the operation performed by an inverter is a negation of the input. If the input configuration of the charges is '1', then the output configuration should match the '0'. The operator implemented in FCN is a particular scheme of containers. The flow of the operation can be described in three steps. Fig. 2.4 shows the fixed input container assuming the logic value of '1'. The fixed input is supposed to not subject to the following containers' forces.

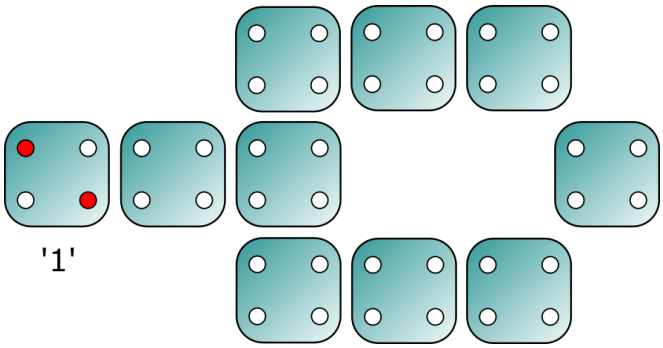


Figure 2.4: FCN inverter: step I

As shown in fig. 2.5, the inverter structure splits the information in two branches. Once it is arrived at the tail, the displacement of the last container allows for a repulsion which leads the output's diagonal to adjust in the opposite direction (compared to diagonal direction of branches data).
Finally, fig. 2.6 reports the logic result of the operation: an inversion has occurred. During all this process, charges acted according to their most stable configuration.

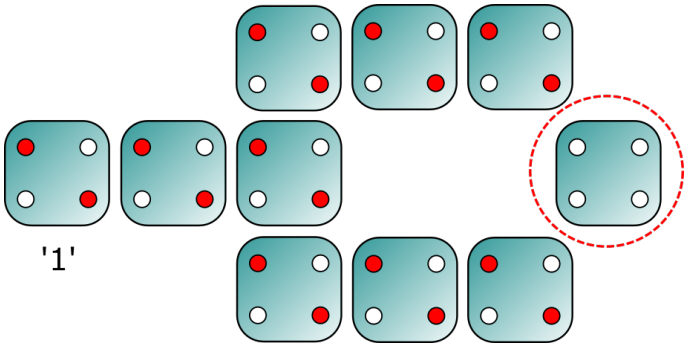


Figure 2.5: FCN inverter: step II

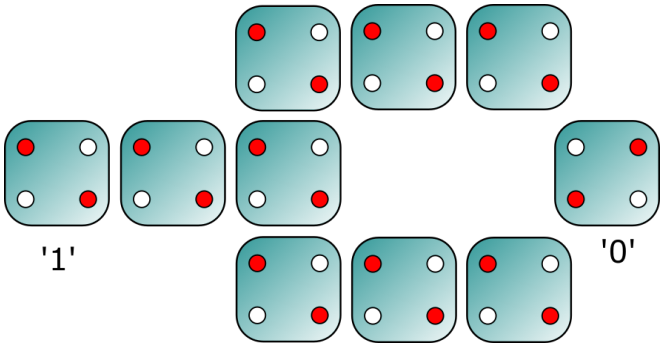


Figure 2.6: FCN inverter: step III

2.3.2 Majority voter

It is a logic operator that takes three logic values as inputs, and provides as a result the logic value "winning" from the majority of the inputs. For example, input values of {0,0,0} would give 0 as result, and input values of {0,1,0} also. The order of inputs should not affect the correctness of operations. The flow of the operation is described in three steps. Fig. 2.7 shows the FCN geometry for majority voter. Two inputs have logic state '1' while the third has '0'.
The container in the middle will arrange his charges according to the strongest electrostatic force. The sum of electric fields will favor the central container to assume the value of the majority of the inputs. Fig. 2.8 shows in orange the most advantaged arrangement of charges inside the central container.
In fig. 2.9 the information has arrived to last container. The last container serves as a "handle" to bring the logic state away from the core of the electric fields. This last step allows the logic result to be propagated and used for further operations.

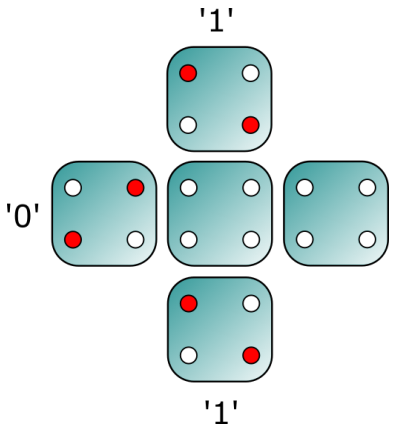


Figure 2.7: FCN majority voter: step I

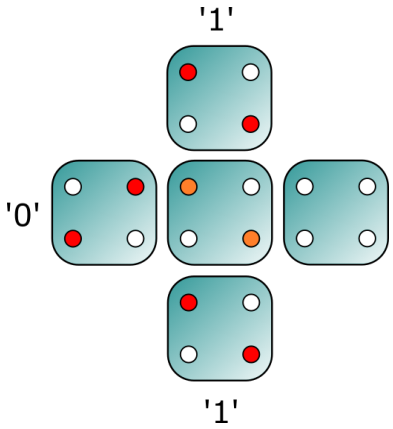


Figure 2.8: FCN majority voter: step II

2.4 Counter-propagation

Coulomb force generated by two equally charged points is a bi-directional phenomenon. This means that, without additional mechanisms, the direction of propagation is not know *a priori*: the force exercised by outputs on the inputs is the same as the one exercised by inputs on outputs. Fig. 2.10 shows how an output could also become an input and lead to an unpredictable behaviour of the wire.

2.5 Clocked FCN

In order to avoid an unreliable direction of propagation, the concept of *FCN clock* has been introduced. Just like, instandard CMOS technology, the clock enables a latch to hold the data or be transparent to it, the FCN clock resets the container and, when it is released, allows the

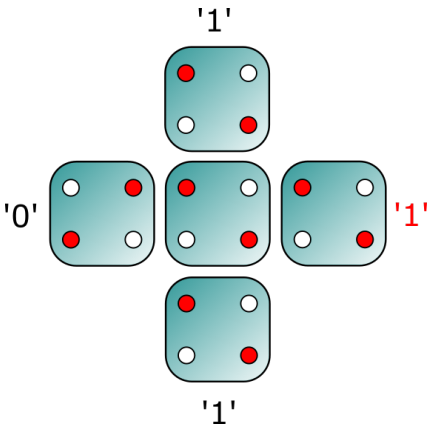


Figure 2.9: FCN majority voter: step III

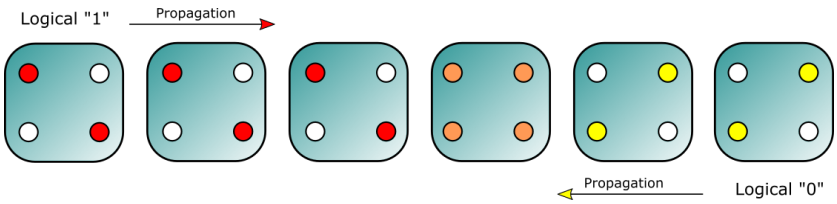


Figure 2.10: FCN wire experiencing a counter-propagation

container to sample the new data. The clocked container is represented with two additional dots, as fig. 2.11 shows.

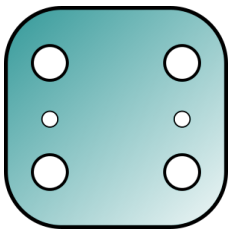


Figure 2.11: Additional dots in clocked FCN container

2.5.1 Hold and reset states

The introduction of a clock adds another functional state to the container: the *reset state*. While on reset state, the container is transparent to the electric field of its neighbors: the charge goes into the smaller holes and becomes inert. When the clock is switched, the charge is pushed

again into its own diagonal-equilibrium configuration. Backward, the old functional state is called *hold state*. The states are represented in fig. 2.12.

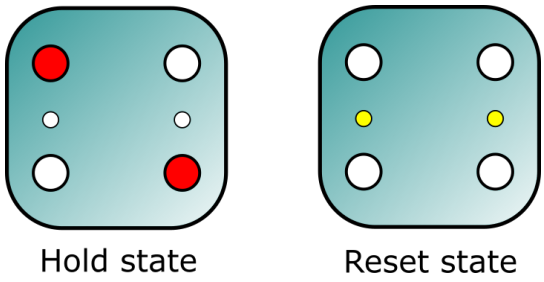


Figure 2.12: FCN hold and reset states

2.5.2 Clocked propagation

The clock hold/reset mechanism enables a guided propagation, which is no more subject to counter-propagation. This is possible due to the following assumption: when transitioning from *reset state* to *hold* state, the new equilibrium configuration of the container is influenced by the pre-existing electric field of neighbours containers. Fig. 2.13 shows how the clock signal guides the logic propagation. Note that the '1' and '0' values of clock are purely by way of example: the mapping between high/low clock and hold/reset states depend on the physical implementation of the FCN concept.

2.6 Possible FCN implementations

The FCN principle of operation is generic, and is suitable to different physical implementations. Some of the devices assumed suitable for the implementations are:

- Metallic Single-electron transistor (SET)
- Silicon heterostructure
- Nanomagnets
- Molecular

Note that the list is purely by way of example, and not intended exhaustive. Among these possibilities, the *molecular* implementation has been chosen and it is the subject of this work.

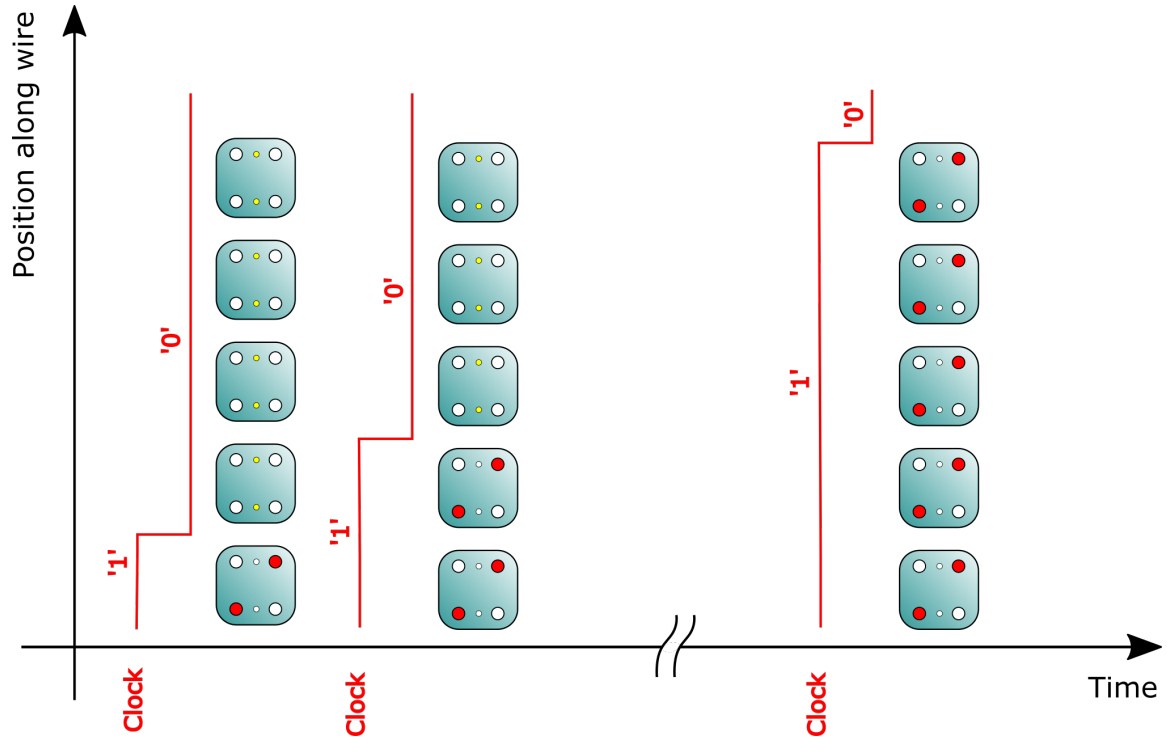


Figure 2.13: FCN clocked propagation

Chapter 3

Molecular FCN: a starting point

The FCN concept is an attractive computing approach, but in to be realized it needs a proper device-level implementation. A possible implementation takes use of already existing stand alone charge containers, which nature has provided: molecules. A molecule has already a predefined electrons cloud. If the molecule is imagined as a container, the displacement of the electron cloud (concerning its equilibrium position) matches the concept of transferring the charge from a dot to another. Therefore, Molecular FCN is introduced (also known in literature as Molecular Quantum-Dot Cellular Automata, or simply MQCA). Since they allow for faster frequencies and lower power, molecules are among the most promising devices suitable for FCN [3].

3.1 The molecular container

Molecules are, by definition, neutral groups of atoms. Their lack of electrical charge is not in contradiction with FCN containers: according to the "dots" concept, the container is globally neutral but only locally charged. However, an oxidized version of the molecule can obtain better performances [3]. For the molecule to be correctly mapped to a *clocked FCN container*, six dots are required. Yet, due to the geometrical pattern generated by the equilibrium configurations, a single container can be thought of as the union of two "sticks": each *stick* incorporates three dots. Within a stick, a thin tunnel connecting the dots is represented. It models the concept of charge transfer between the dot. When an electrostatic force pushes the charge towards the opposite (empty) dot, it has a preferred lane, which depends on the technological implementation. This representation is shown in fig. 3.1,

3.1.1 Conceptual molecules

The cut of a container into two sticks does not change the global behaviour. However, this last representation grants an easier molecular implementation. A generic molecular structure able to match "stick" scheme is presented in fig. 3.2. The molecule that implements the stick requires three separate areas that assume the role of the dots.

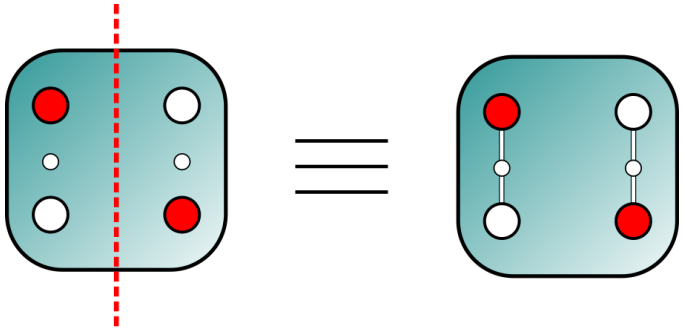


Figure 3.1: Representation of charge container as the union of two *sticks*

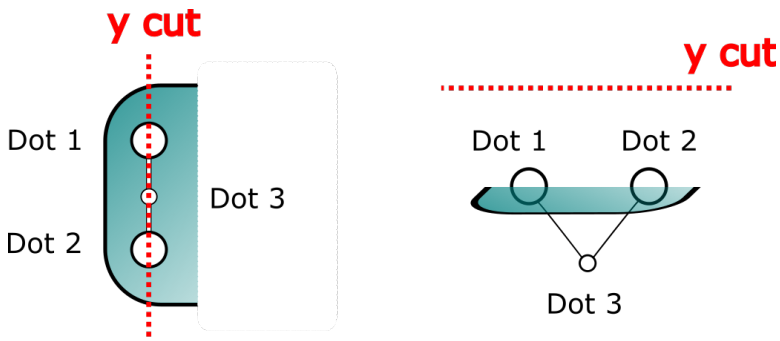


Figure 3.2: Possible position of molecular dots in a candidate molecule

By virue of prior division, two molecules are necessary to build the container. Molecules have to be organized one in front of the other. Fig. 3.3 show a 3D and a top view of the molecular container.

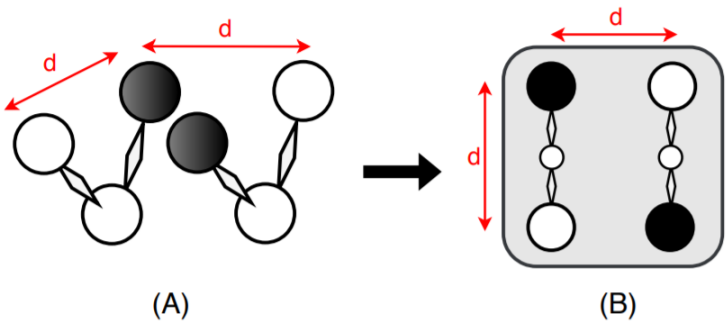


Figure 3.3: 3D view and top view of the molecular container [4]

3.1.2 Bisferrocene for FCN

When thinking of a molecule as a charge container, there is a key requirement: molecule should emulate the concept of dots. A zone inside the molecule that is suitable to accumulate or disperse electron's charge, is a zone where *reduxion-ossidation* occurs, also known as **redox** center [11]. Among the possible molecules, one of the most suitable for this purpose is the

6-[3, 6-bis(1-ethylferrocen)-9H-carbazol-9-yl]-6-hezan1thiol

also called *Bisferrocene* [11], whose structure is presented in fig. 3.4.

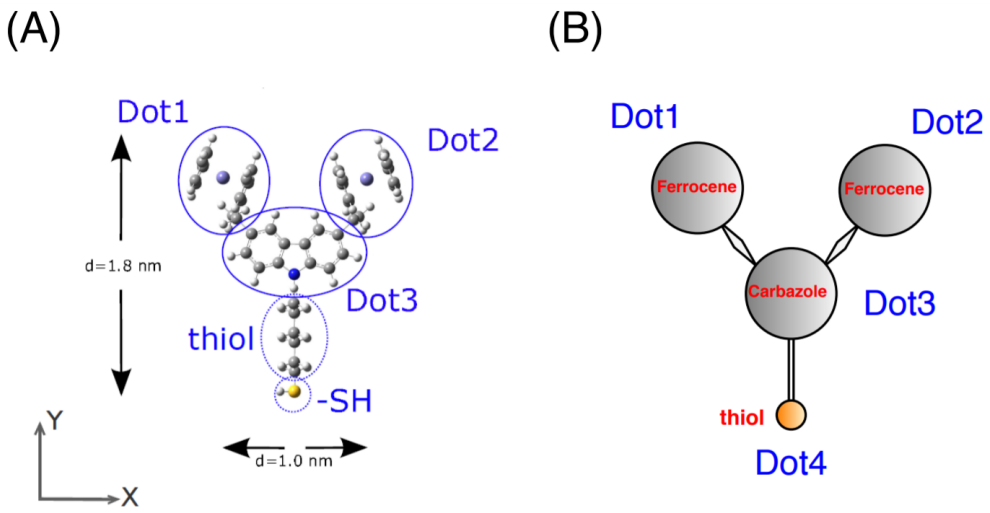


Figure 3.4: Bis-Ferrocene molecular structure and its schematic structure [4]

3.1.3 Redox centers

The former molecule matches the key requirements to implement a FCN container. It has three redox centers (Dot1, Dot2 and Dot3) with a spatial distribution that reminds the arrangement of dots in the conceptual molecule conjectured. The *thiol* group allows the molecule to be anchored to a gold substrate. If the molecule is ionized, its charge is easily displaced by an electric field [11]. When subject to a proper electric field, the charge layout acquires the shape displayed in fig. 3.5. The corresponding schematic representation is shown for each charge localization in fig. 3.6.
The render shows charge localization while in reset state (A), and in binary logic configurations (B, C).

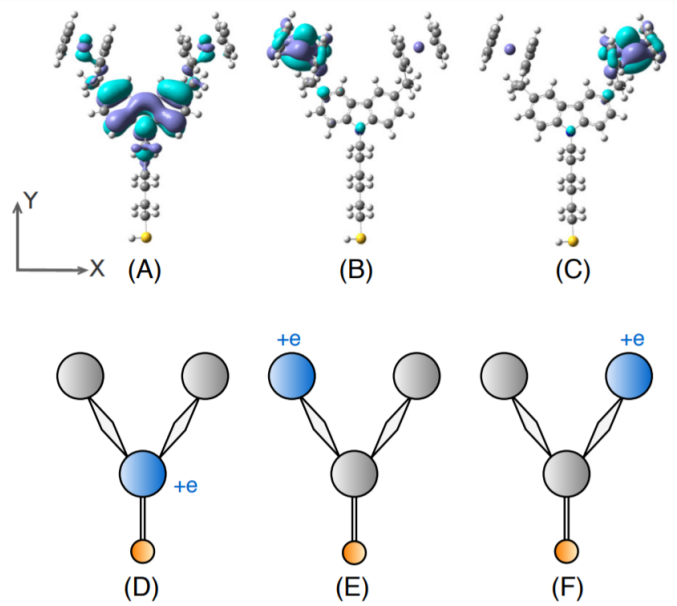


Figure 3.5: Bis-Ferrocene charge layout and positive charge displacement [4]

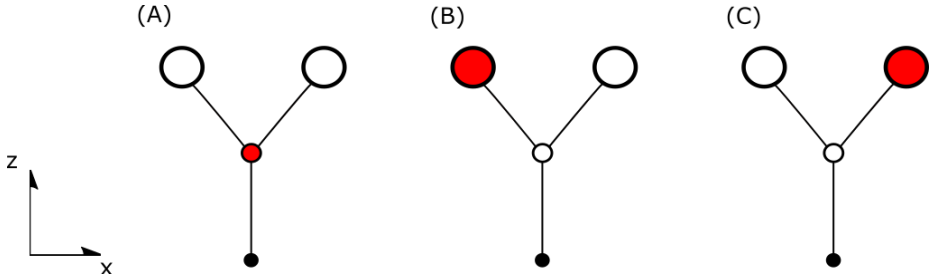


Figure 3.6: Bis-Ferrocene schematic structure and positive charge displacement

3.2 Applying an electric field

The displacement of the charge from its equilibrium condition requires an external electric field. The geometry suggests that the influence of the electric field shall arrive from two distinct directions. A horizontal electric field could change the charge distribution in the upper dots (fig. 3.5 B, C) and so switch the logic state. On the other hand, a vertical electric field could be exploited to implement the Clocked-FCN concept, by pushing the charge in the region suitable for reset state (fig. 3.5 A).

3.2.1 Horizontal electric field

The basic principle of FCN relies on the proximity of charges to propagate a particular dots configuration. Even for the molecular FCN approach, then, charges should be able to generate an horizontal electric field. Fig. 3.7 shows how the field generated by a charge localization is diffused into nearby molecules.

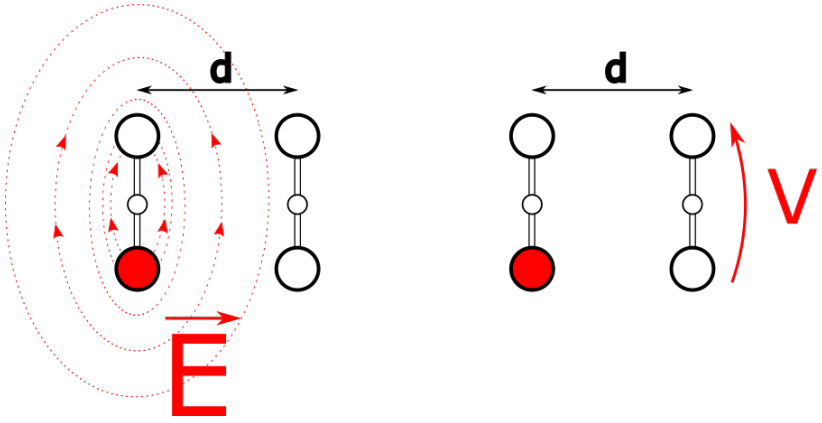


Figure 3.7: Horizontal electric field can be modelled as an input voltage on the second molecule

The effect of the field generated by the *driver* molecule can be modelled with an equivalent voltage applied on the second molecule [10]. The horizontal field can, therefore, be seen as signal that propagates on a molecular FCN wire: this reminds the (CMOS) concept of a voltage signal that crosses the combinatorial paths. In fact, simulations have demonstrated output characteristics (Vout / Vin). The value of the equivalent voltage is strongly conditioned by distance **d** between two consecutive molecules.

3.2.2 Vertical electric field

The horizontal electric field represents how the true information propagates in FCN. Nevertheless, as in order to have a *guided* propagation, a clock signal should force the charge to be localized into the central dots. By looking at the molecular structure, this means that the electric field should be vertical, hence the charges can be pushed to the bottom electrode. Fig. 3.8 shows how an the previous practice can be achieved: molecules are placed between two electrodes. Different voltages applied to electrodes lead to distinctive container states, already discussed in chapter 2.

3.3 Molecular FCN Layout

In this section the earlier discussed notions are exploited for a possible molecular layout. The purpose of the layout is to describe an arrangement of electrodes and molecules capable to actualize the FCN method.

3.3.1 Electrodes structure

Vertical and horizontal electric field are crucial to accomplish the guided FCN operations. Horizontal electric field arises from a well-defined placement of molecules. Vertical electric field

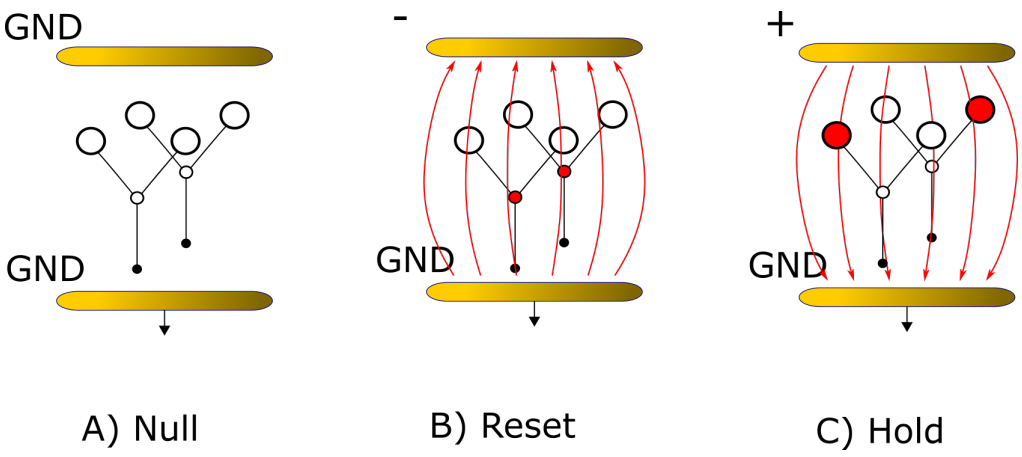


Figure 3.8: Effect of a vertical electric field on the negative charge localization

originates by external electrodes. For everything to work properly, electrodes have to be placed accordingly. Fig. 3.9 report a section of the possible electrodes scheme. On a dielectric layer, a trench is dig. Inside the trench think gold nanowire is deposited. By virtue of the thiol group, molecules can adhere to the gold nanowire. On the top of the trench, the two placed clock electrodes are capable to generate the vertical electric field (essential for guided clock operations).

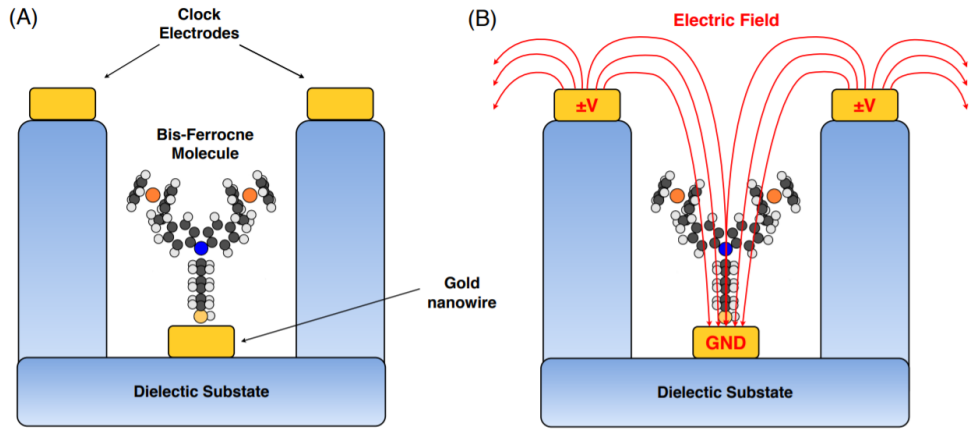


Figure 3.9: Section of a possible electrodes layout [4] according to FCN paradigm

3.3.2 The molecular wire

The section in the previous scheme can be longitudinally extended to form a molecular wire. In order for two subsequent molecules to match an FCN container, the length of the clock electrodes should not exceed the distance between the two molecules. However, manufacturing

of clock electrodes don't allow for a very short electrodes size. Nevertheless, pairs of molecules can be added to a container without changing its logic properties. Clock zones will then define the container's boundaries, as shown in fig. 3.10.

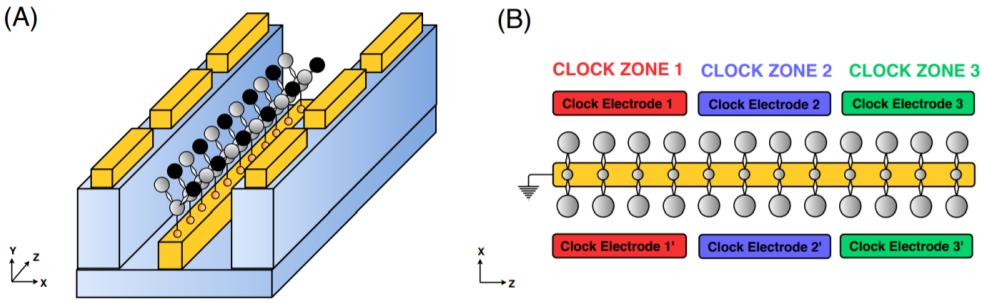


Figure 3.10: Layout of an FCN molecular wire [4]

3.3.3 Half adder: Molecular FCN implementation

To better understand what are the implications of the molecular FCN implementation and the previous layout, a half-adder design has been reported. Fig. 3.11 shows a top-view simulation of an inverter. Fig 3.12 displays an aerial view of the molecular wires/logic gates necessary to perform the operation Half-Adder operation. Fig. 3.13 reports the top-view scheme of the same circuit.

3.4 Manufacturing issues

One of most promising advantages of the molecular implementation of FCN is supposed to be the manufacturing. Molecule deposition is more "natural" than carving a 3D FinFET. A possible synthetization technique for Bisferrocene could exploit the following steps:

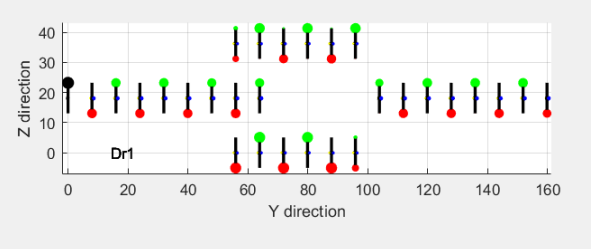


Figure 3.11: Top view of an inverter implemented in Molecular FCN

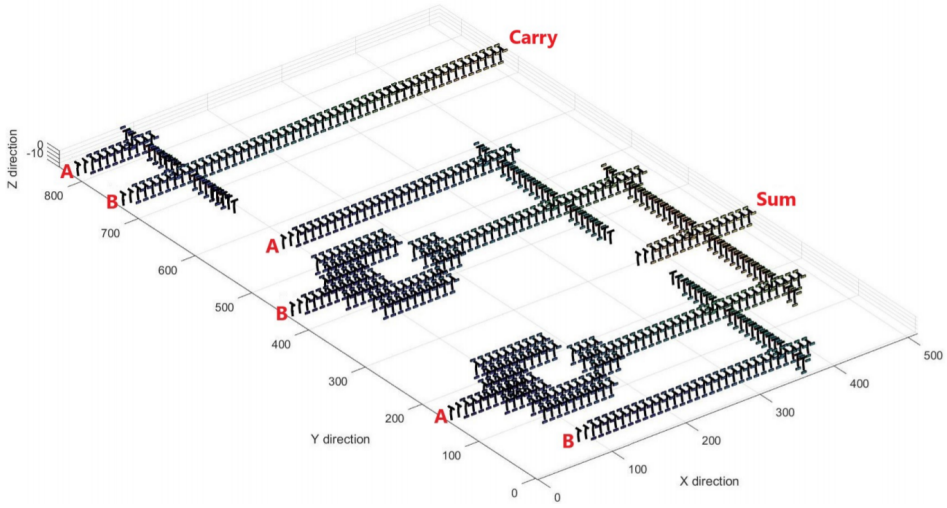


Figure 3.12: 3D view of an Half-Adder synthesized in molecular FCN [4]

a. Shape the gold electrodes

b. Dip the structure into a thiols solution. Thiols create bonds with gold.

c. Dip the structure into a bisferrocene solution. Bisferrocene group will bond with thiols: depending on their concentration and hindrance, the molecular distance will be affected.

However, molecular FCN fabrication has to deal with possible obstacles. Two issues have been reported here: the shaping of the wires, and the possible logic inversion due to odd molecules placement.

3.4.1 Shaping of wires

An easy manufacturing process should possibly rely on constructing standard repetitive patterns, rather than trimming each logic gate in a different manner. From this point of view, the molecular FCN implementation which has been described, doesn't meet the criteria. Self-Assembled Monolayers allow for a uniform distribution of molecules, on which molecular wires/operators have to be drawn. Even tough it is possible to introduce patterned features in a Self-Assembled Monolayer, they could not meet the shaping accuracy required for Molecular FCN: it is enough the presence of a single molecule in excess to create a logic inversion on the data.

3.4.2 Logic inversion due to odd molecules placement

The use of the peculiar Bisferrocene molecule forces the latter to be mapped into a 3-dots *stick*, rather than a 6-dots container [11]. This implementation sets a constrain on the periodicity of

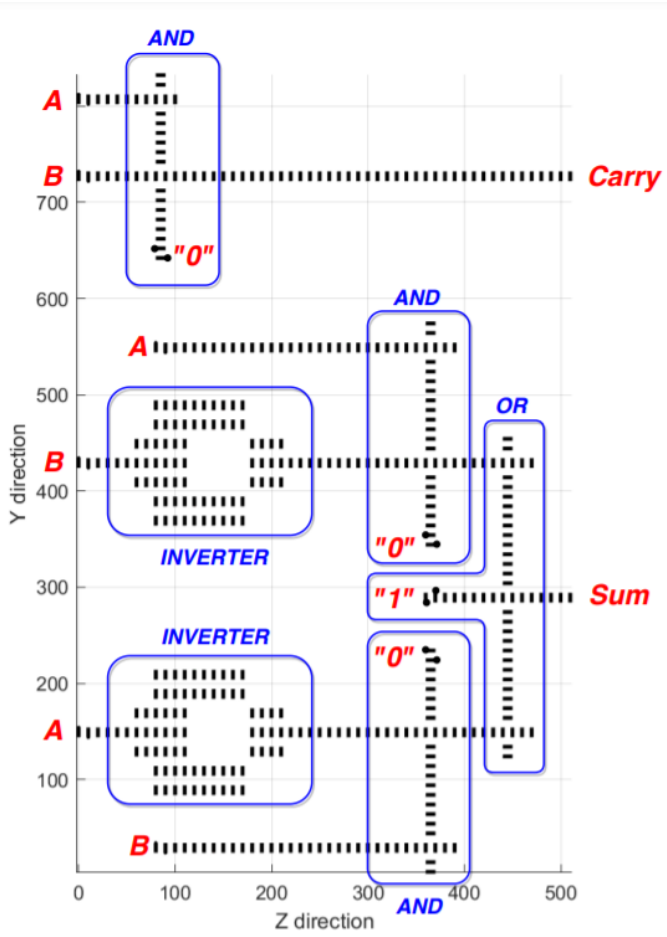


Figure 3.13: Top-view of an Half-Adder synthesized in molecular FCN [4]

the molecular placement. Since the *stick* acts like an inverter, it must be placed in pairs. An odd number of molecules (sticks) would perform an unwanted and not predicted inversion of the logic data, as shown in fig. 3.14.

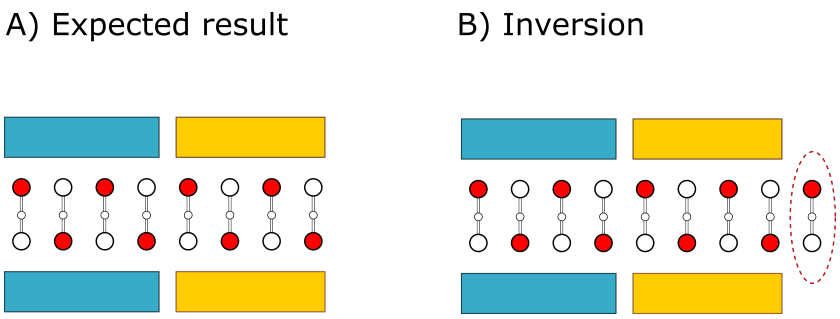


Figure 3.14: A spare molecule leads to an inversion of data

Chapter 4

A new FCN concept

The molecular implementation of Field Coupled Nanocomputing could bring noticeable advantages on current silicon technologies in terms of speed and power [3]. Unfortunately, noticeable manufacturing effort is required in order to shape the molecular wires. A too complicated fabrication could delay the prototyping of FCN circuits and would preclude the FCN paradigm from the podium of "Beyond CMOS" alternatives. Here comes the need of a new FCN concept, that aims to overcome the manufacturing difficulties while keeping (and possibly extending) its advantages over CMOS technology.

4.1 Possible scenarios

In this section two novel ideas are discussed. Both have been conceived with molecules in mind: the intent was to increase the physical feasibility of FCN's molecular variant. However, they are general concepts, and due to their abstraction they are suitable to any physical implementation.

4.1.1 Vertical Stacked FCN

Vertical Stacked FCN exploits the vertical spatial trajectory rather than the horizontal direction. This proposal defines a completely different assembling process.

Concept

Electrostatic fields originated by a point electric charge are isotropic. In principle, then, it is possible to organize the spatial allocation of dots in a different scheme, provided that propagation and logic processing are still correct. For this purpose, the fig. 4.1 presents a compliant arrangement.
According to the proposal in fig. 4.1, containers are stacked one on top of the other. From the point of view of the Coulombian electrostatic repulsion there is no change: the portrait of dots in the bottom container is mirrored in the upper container. Containers are alternatively stacked in

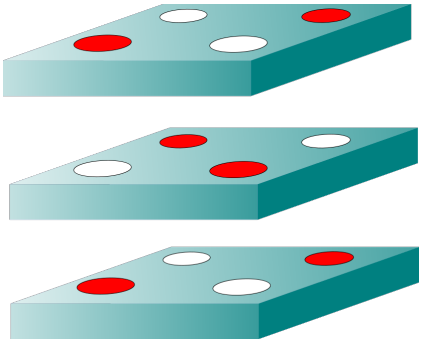


Figure 4.1: Representation of a vertical FCN

the vertical direction, exactly how "sticks" were alternating in the horizontal direction in classic FCN.

Manufacturing

The key point of Vertical Stacked FCN would be the peculiar fabrication method. Supporting the molecular scope, *Atomic Layer Deposition (ALD)* grants the capability to place the exact amount of layers. ALD is a thin-film deposition technique that involves sequential placement of atomic layers. Each layer bonds to the subsequent until a first molecular FCN layer is completed. ALD process is shown in fig. 4.2

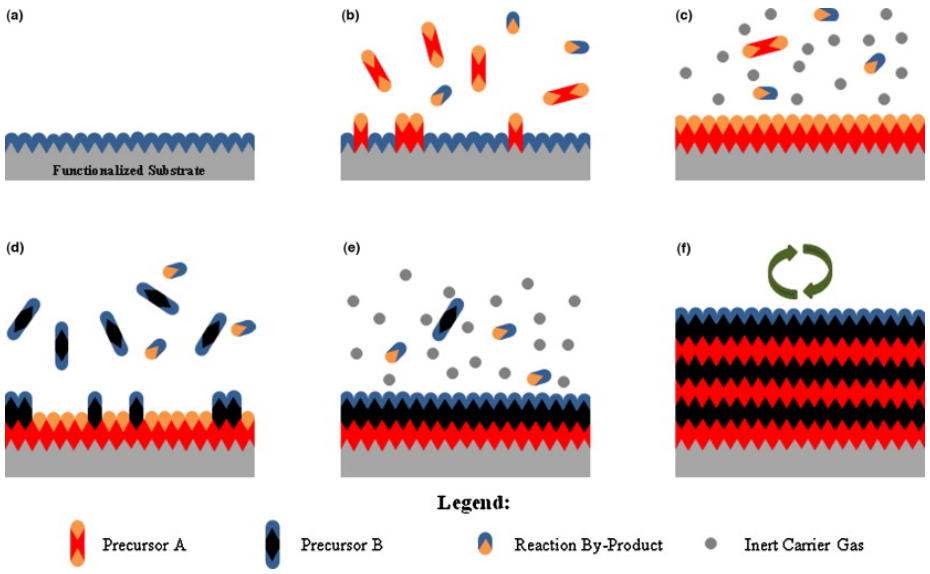


Figure 4.2: Common ALD procedure leads to alternate layers [5]

Between the first and the second FCN layers a dielectric atomic structure could be exploited,

allowing the field lines to propagate in the vertical direction with low perturbation. Because this method forecasts the explicit number of layers that will be present at the end of the process, this technique solves the risk of an extra inverter generated by odd molecules placement, as discussed in subsection 3.4.2.

4.1.2 Reconfigurable FCN

Classic FCN makes use of well-shaped distribution of containers, depending on the desired action (propagation along wire, logic operations). Nevertheless, without a clock signal that guides the action, counter-propagation can occur. Provided that a clock signal is necessary in any case, and since the signal selectively turns on the required containers, there is no actual need of placing the containers in a speficic scheme. They can be placed everywhere and then activated by the clock according to the needs. The Reconfigurable FCN approach exploits for the first time a **regular** grid of containers rather than a custom arrangement. The FCN containers are only selectively activated, and the activation order resembles the geometric scheme of the classic FCN. Paths will be shaped through electrical signals (or other physical observables), allowing, in addition to logic operations and propagation, a totally new approach for data manipulation.

Concept

A logic operator implemented in classic FCN has a pre-defined geometrical scheme. The space that is not occupied by a FCN container can also be "empty". According to the new FCN structure, it is filled by **turned-off containers**, as fig. 4.3 displays.

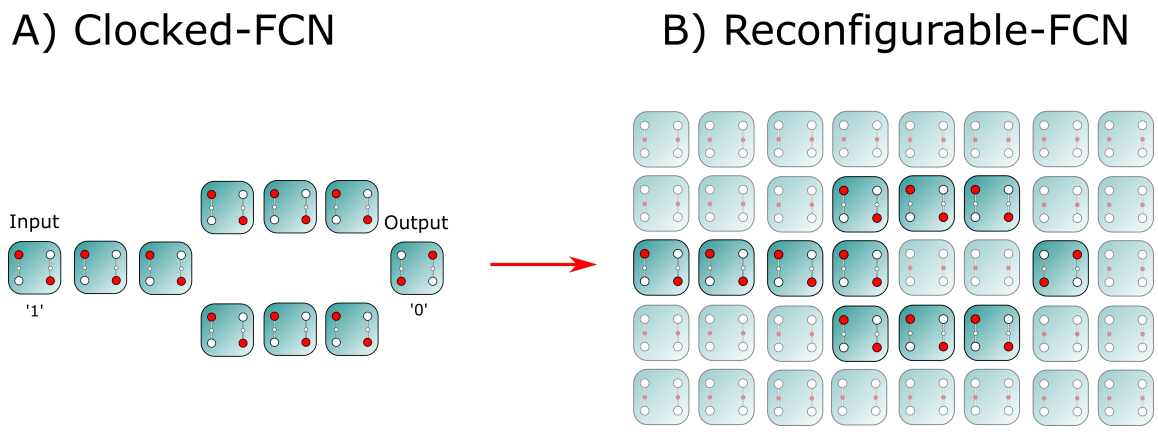


Figure 4.3: RFCN: unused space is filled by empty dots

This approach does not require further signals (other than the clock) to be handled: the *enable* of clocked-FCN can be reused here to activate only a subset of containers. Provided the ability to properly coordinate the clock signals, it is possible to reconfigure the logic run-time: here comes the name choince of Reconfigurable FCN,

Manufacturing

The manufacturing of Reconfigurable FCN is benchmarked on its molecular implementation. The molecular fabrication process takes advantage of *Self-Assembled Monolayers (SAM)*. Instead of shaping the single wire, a very large ordered domain of molecules is simultaneously assembled over a surface. According to this scenario, this new Reconfigurable Molecular FCN concept should rely on a single, uniform matrix of molecules, rather than many different, custom placements. A Molecular SAM is represented in fig. 4.4.

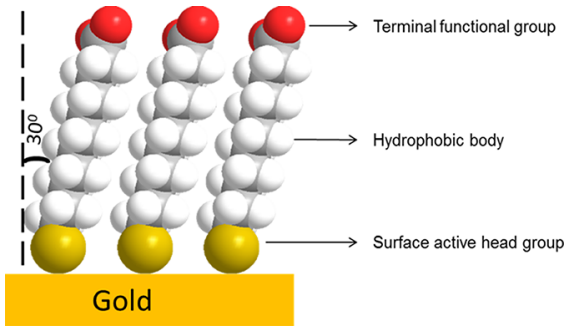


Figure 4.4: Molecules deposited by SAM [6]

After placement, molecules are conceptually grouped into *molecular blocks*, that act like a container. Each block is subject to an external enable signal coming from a clock electrode: it will be the key element to define the block (container) boundaries.

4.2 Choice of the concept

The two possibilities examined in the previous chapter provide a more straightforward manufacturing, in both cases of SAM and ALD. For this work, the Reconfigurable FCN is chosen to be further developed and discussed.

Chapter 5

Reconfigurable Field-Coupled Nanocomputing

Reconfigurable FCN extends the concept of Field Coupled Nanocomputing. In fact, it grants new possibilities of logic and data manipulation. Among the other features, the opportunity of reconfigurability is a key element, and has been chosen to label the new approach. Reconfigurability will be discussed in next sections.

5.1 Containers'ocean

Reconfigurable FCN *(RFCN)* arises from the idea of having a uniform large grid of containers, such that any technological implementation will possibly take advantage of a regular large area patterning, rather than a nano-scale shaping. Fig. 6.4 shows an "ocean" of containers extending in both horizontal directions. The boundaries of the plain are determined by the designer. As this amount of containers will be placed on the same layer, the containers' ocean will be referred as the *Containers Common Layer (CCL)*.

5.1.1 A starting point: clocked FCN

Considering that simple periodic placement of containers does not permit the task differentiation, something else is needed to "specialize" the areas and instruct them to perform a specific action. The idea of a *clock* that selectively enables containers is therefore essential also in RFCN. According to the former clock convention, it is only when enalbed that the container can interact with neighbors. Fig. 5.3 provides a an example of a subset of activated containers surrounded by inactivated containers.
The concept of "enabled" and "inactivated" can be associated, respectively, to the "hold state" and "reset state" discussed in section 2.5. It is possible to see in fig. 5.3 the graphical representation chosen for hold and reset states.

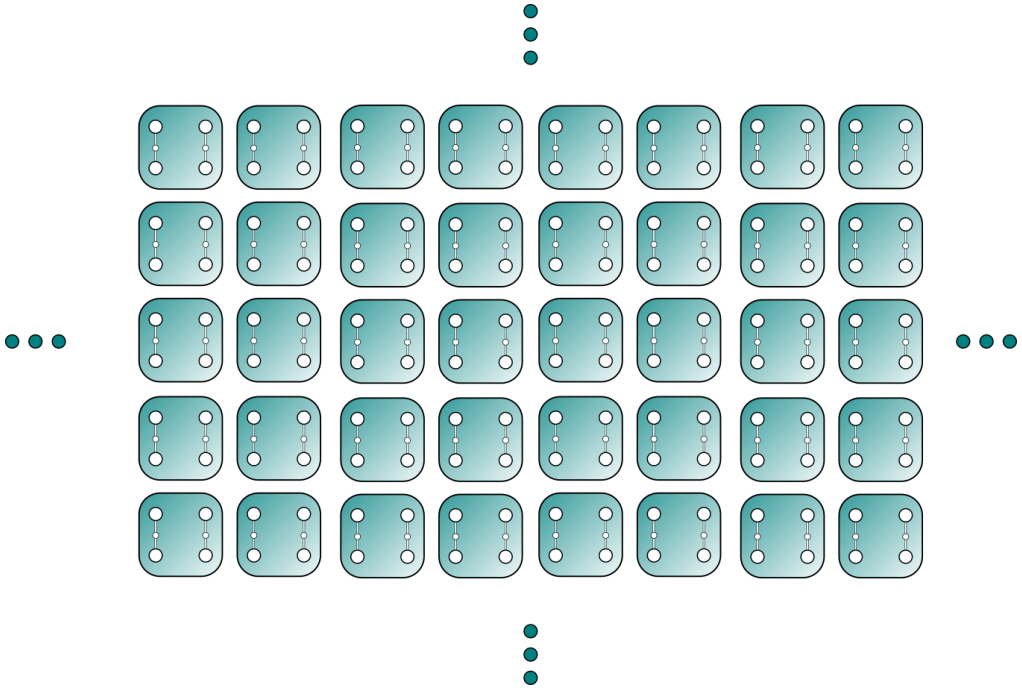


Figure 5.1: The ocean of containers

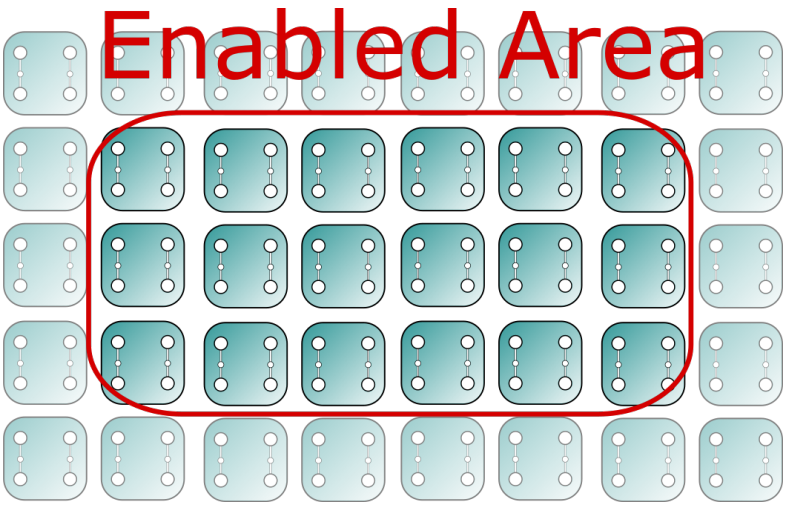


Figure 5.2: Effect of the clock on a subset of containers

5.1.2 Logic propagation

The simplest, straightforward task that RFCN can achieve is the propagation of the logic information. Fig. 5.4 presents a propagation scheme for RFCN. The picture on the left shows the turning-on sequence of the clock, that reflects into the gradual activation of containers, which

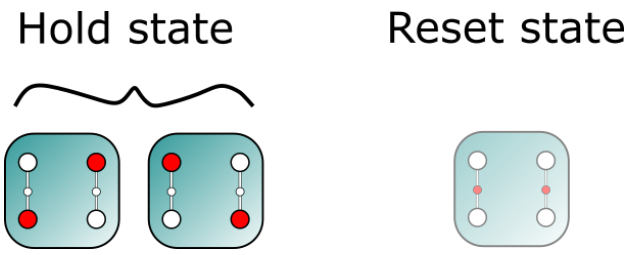


Figure 5.3: Identification of the hold and reset states

is instead shown on the right. At first sight the change with respect to classic FCN is minimal. The only distinction is the presence of the surrounding turned-off data containers.

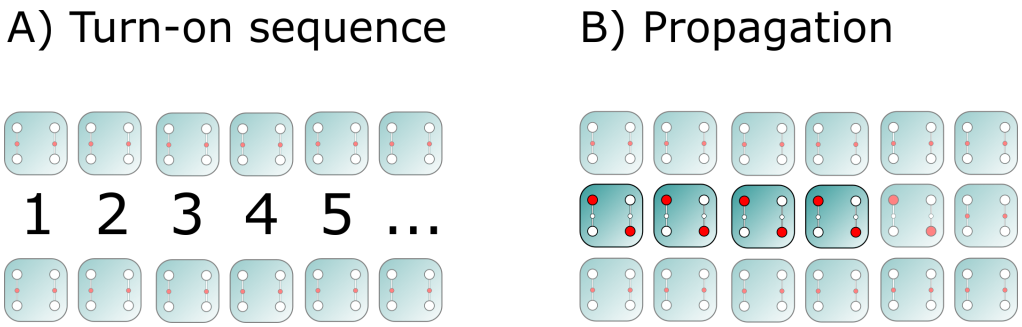


Figure 5.4: Propagation of information along a wire

5.1.3 Logic operations

Also the logical operators' schemes can be inherited from FCN. The type of logic operation performed is a direct consequence of the geometrical arrangement of the containers. RFCN indeed extends the functional behaviour of the containers, nevertheless, it doesn't touch the relative position of containers. Therefore, logic operators' geometries can be re-used.

Majority voter

The RFCN majority voter acquires the exact form of its classical FCN version. Also the evolution of clock-zones is copied. Fig. 5.5 displays the scheduled evolution of the clock, while fig. 5.6 shows the actual progression of turned-on zones.

Inverter

Inverter has two possible implementations in RFCN. The first one re-traces the classic FCN implementation: the geometry and clock activation sequence is kept. However, there is a more

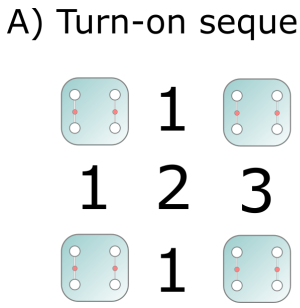


Figure 5.5: Scheduled sequence of clock signal's activation

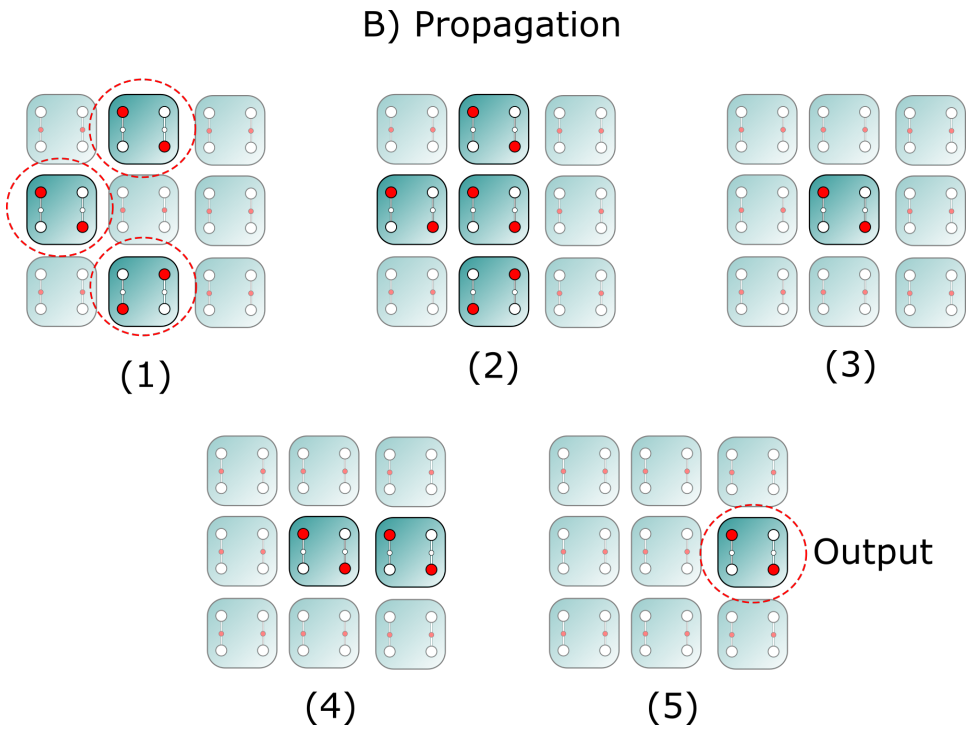


Figure 5.6: Animation frames of the calculation

compact solution. Figures 5.7 and 5.8 present an inverter that occupies the same area as the majority voter, and avoids the long-tail of the classic inverters (fig. 2.4). The operation is obtained through a **spatial reuse** of the containers in different time stamps. This is a first example of a "local" reconfigurability: inside the area delimited by logic operator, the same containers are re-used to accomplish the inversion operation.

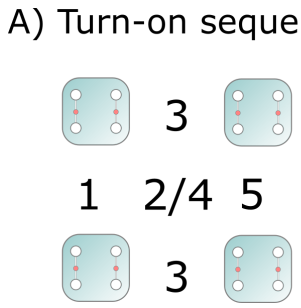


Figure 5.7: Scheduled sequence of clock signal's activation

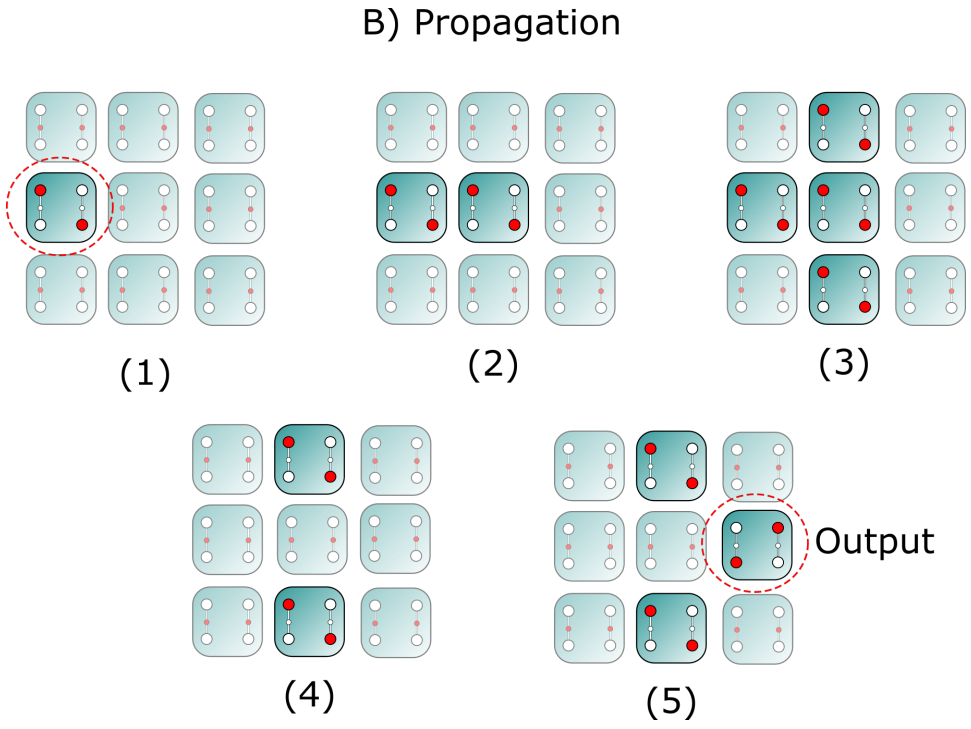


Figure 5.8: Animation frames of the calculation

5.2 Standard-cell concept

From a logic standpoint, majority voter and inverter are enough to guarantee logic completeness [12]. Taking into account that these previous operations can be performed within a 3x3 group of containers, a **standard-cell** concept can be introduced: a pre-defined, standard, group of containers that can be used as building block for more complex designs. The RFCN standard-cell idea is shown in figure 5.9. As can be seen from the picture, the number of containers that characterize a standard-cell is 9.

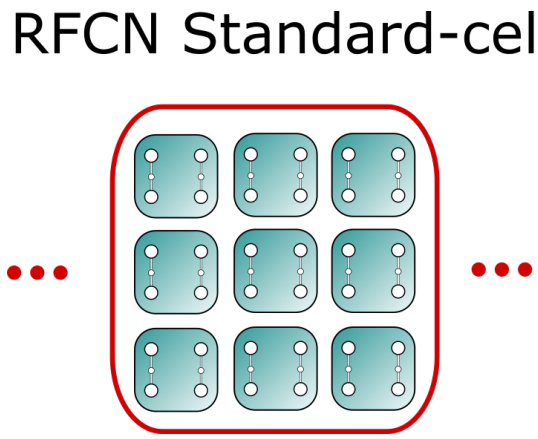


Figure 5.9: Definition of an RFCN Standard-cell.

5.2.1 Functional zones of the standard-cell

By looking at the evolution of turned-on zones during an inverter operation, can be observed how only five out of the nine containers are switched. This circumstance also occurs during majority voter operation (fig. 5.6). A summary of the switchable containers inside the standard-cell is graphically represented in 5.10

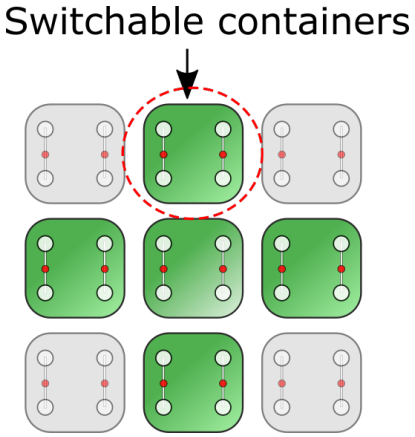


Figure 5.10: Scheme of the containers that are selectively enabled during inverter and majority voter operations (green).

5.2.2 Reset zones of the standard-cell

As it can be deduced from a complementary point of view, the containers that don't switch during an operation are permanently kept in a reset state, as shown in fig. 5.11.

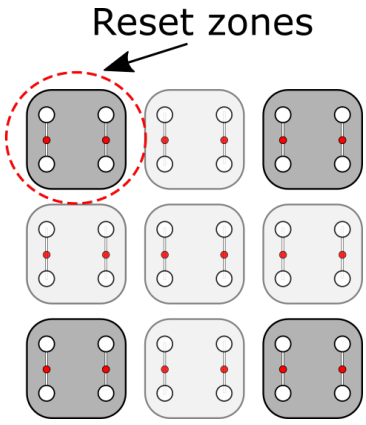


Figure 5.11: Scheme of the containers that are never enabled during enabled during inverter or majority voter operations (dark grey).

5.3 Clock hierarchy

A key point of Reconfigurable FCN (and any other clocked-FCN scheme) is the modelling of the clocking network and, generally speaking, the timing. In CMOS technology, clock is used to define a time window in which a combinatorial function is executed and stored, usually called "T_{cycle}". In FCN, instead, in addition to define a such time-window, clock should also pay attention to containers' sub-timing. In fact, containers exercise a pre-defined turning-on sequence, which demands a proper timing. This new requirement advices the conceptual re-organization of the clock into two hierarchical levels:

- Micro Clock "μ-CK"
- System Clock "S-CK"

5.3.1 Micro Clock: clocking the standard-cell

Unlike CMOS standard-cells, the RFCN cells require a *guide*, which is here referred as "micro-clock". The *micro-clock* is a "low level" (local) clock: it operates directly within standard cell containers. For the purpose of clarifying the clock mechanism, an inverter operation is taken as example and analysed.

Timing

Each basic logic operation (majority voter, inverter) is in turn divided into smaller *"time frames"*: they are the sub-intervals of the timing window requested by the whole operation. Since they are driven by the micro-clock, *frames* will be labeled as *micro-operations*. A micro operations can be defined as the longest period that occurs before the micro-clock modifies the scheme

of active containers. If N is total number of micro-operations that are performed, the overall operation period can be expressed as:

$$T_{op} = \sum_{i}^{N} T_{\mu op}$$

Fig. 5.12 shows for each frame the evolution of the 9 enable signals (one for each container) of the cell. The number that identifies the signal relates it to the corresponding container.

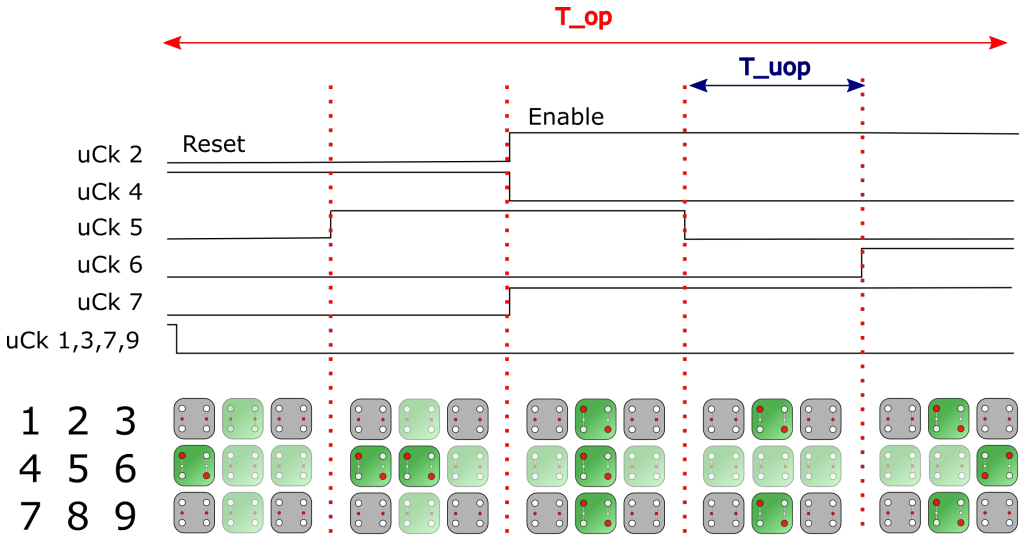


Figure 5.12: Timing diagram showing the time evolution of the micro-clock signals during an inverter operation.

Between a micro-operation and the next one, the scheme of micro-clock signals (enable) changes. Because any real implementation of the already stated signals will not be instantaneous, a settling time will have to be taken into account when changing to a new clock scheme. This is what will define the period $T_{\mu op}$, in addition to a certain slack margin that has to be guaranteed:

$$T_{\mu op} = t_{settling} + t_{margin}$$

Fig. 5.13 shows a micro-clock transition and its relative timing diagram.
As the picture displays, the micro-operations needed for the inverter are 5.

Cell sequencer

Within a cell, the same logical operation will likely be performed many times, in different times stamps. Each time it is executed, the sequence of the micro-clock signals will be repeated. It can then be stored inside a memory, and it can be triggered when demanded. When the operation is supposed to start, the flow of micro-clock signals could be read from a writable memory, called *sequencer* (fig. 5.14).

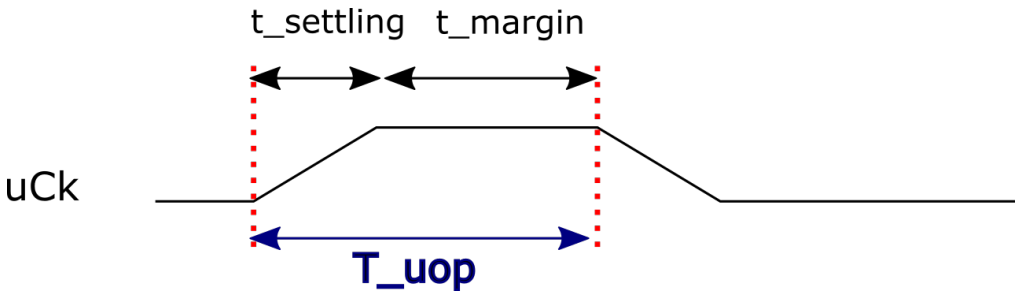


Figure 5.13: Time period of a micro-clock cycle.

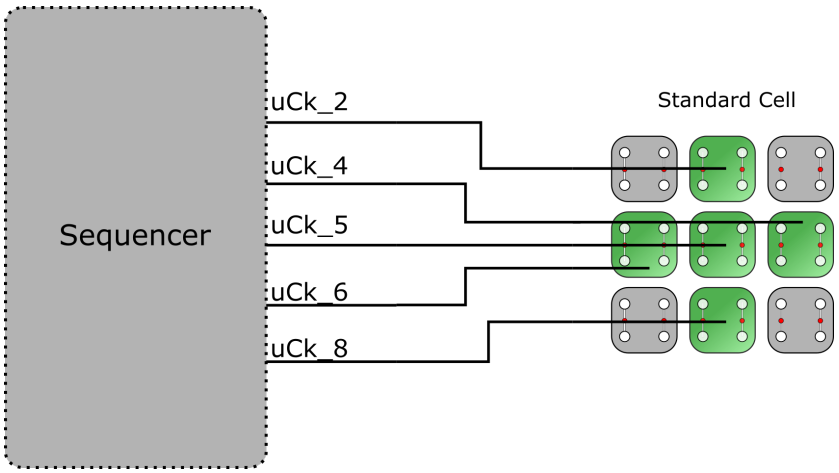


Figure 5.14: Sequencer distributing the micro-clock signals to the standard-cell

5.3.2 System clock: clocking the operations

The micro-clock takes care of the sub-steps that describe the activating sequence of the containers, but the frequency at which the operation is performed is unknown. An external signal is needed to trigger the sequencer. This external signal is represented by the system clock, which is a high-level clock. The latter should resemble the function scheme of CMOS clock: synchronise operations and data dependency. In fact, this clock hierarchy mimics a Hierarchical Control System (HCS), as shown in 5.15.
The only timing constrain on the system clock is its period. The period should be greater than the highest operation duration, to ensure enough time for the logic operation:

$$T_{SCK} > T_{op_{max}}$$

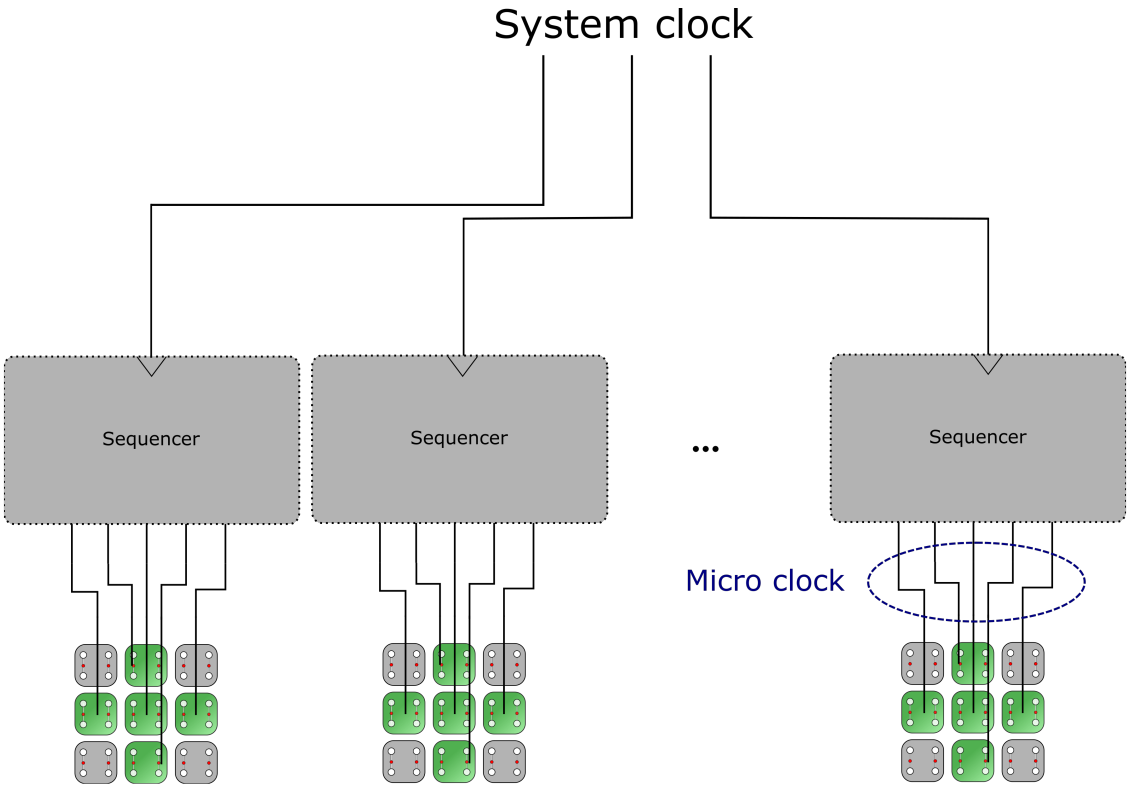


Figure 5.15: System clock and Micro clock under a hierarchical representation

5.4 Logic synthesis

An important link between the new RFCN concept and its utilization in VLSI is represented by the logic synthesis. The news with respect to CMOS is that the prospect of a totally different synthesis. Substantially, it occurs by means of electrical signals (instead of physical patterning).

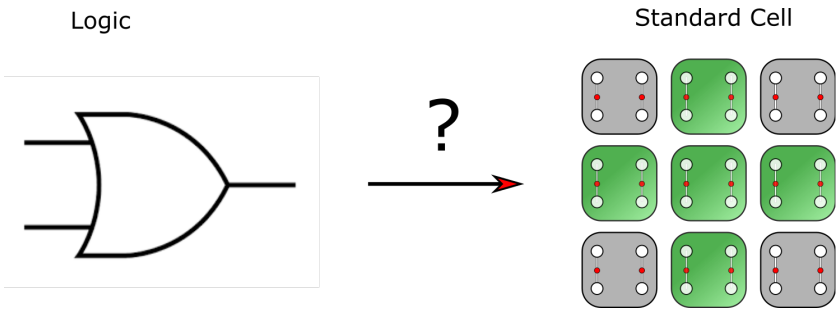


Figure 5.16: Concept of mapping a generic logic gate into RFCN

5.4.1 AND gate

Logically, an AND gate can be built throught a majority voter with one of its inputs set to 0. The structure of an AND gate is shown in fig. 5.17. However, to have a standardized inputs scheme, it is proposed to make all the inputs arrive from the same column coordinate. Fig. 5.18 shows the two additional cells added for this pourpose.

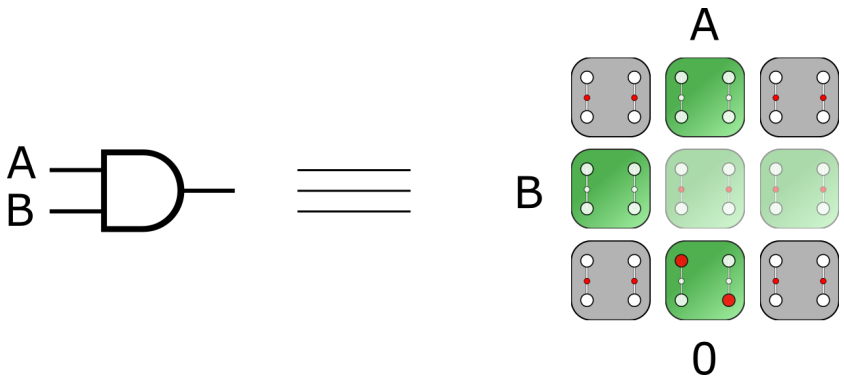


Figure 5.17: AND gate mapped by a single standard-cell

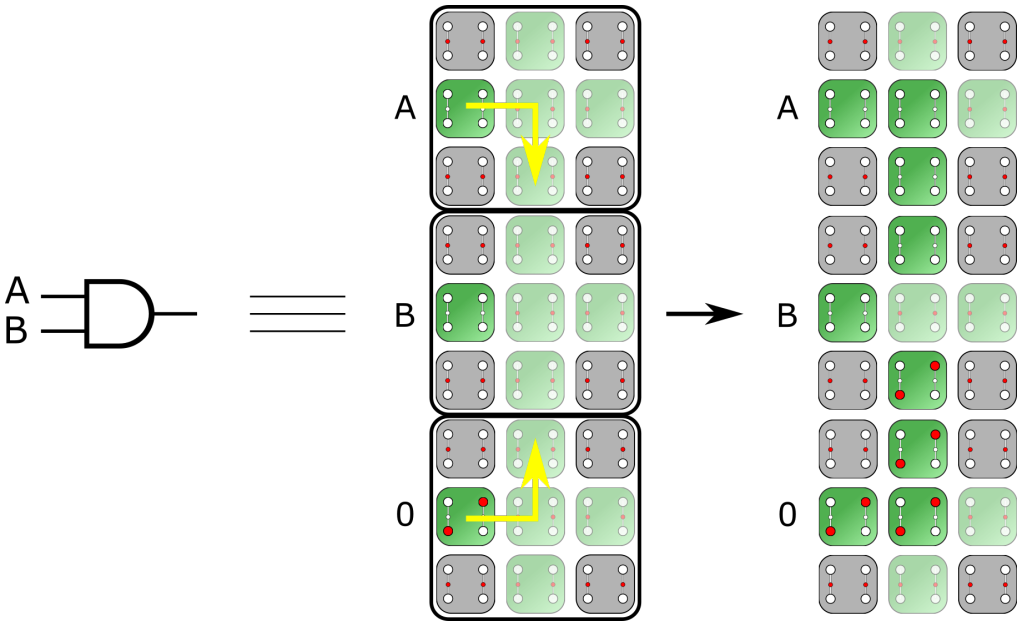


Figure 5.18: Implementation of AND gate with two additional standard-cells for inputs synchronisation

5.4.2 NAND gate

A basic logic function is represented by the NAND: it can be obtained by the union of an inverter and an AND gate. In FCN it is enough to instruct three standard cells to perform the AND operation, and a forth standard cell to perform the inverter. Figure 5.19 shows the relevant standard-cells involved in a NAND operation. Note, however, that the surrounding standard-cells aren't shown for graphic purposes.

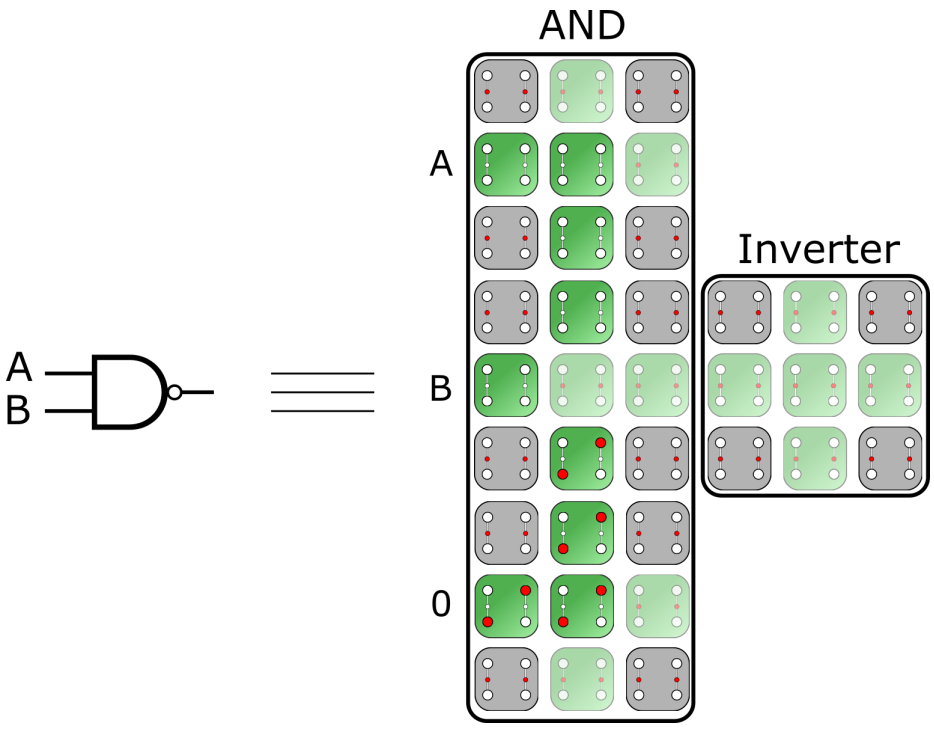


Figure 5.19: NAND gate mapped to RFCN through spatial union of AND and inverter operators

5.4.3 Exclusive-or-gate (XOR)

A widely used logic operation is expressed by the exclusive or gate (XOR). Its translation in the RFCN scheme is a starting point for the new type of synthesis. The structure of a XOR gate is shown in 5.22.

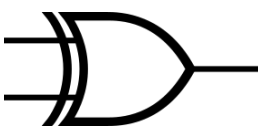


Figure 5.20: Logic scheme of a XOR gate

In case of AND and OR gates mapping, the third input of the corresponding majority voter should be set, respectively, to 0 and 1.

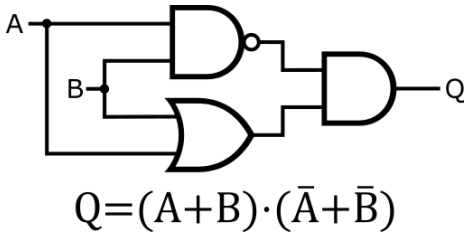


Figure 5.21: XOR structure

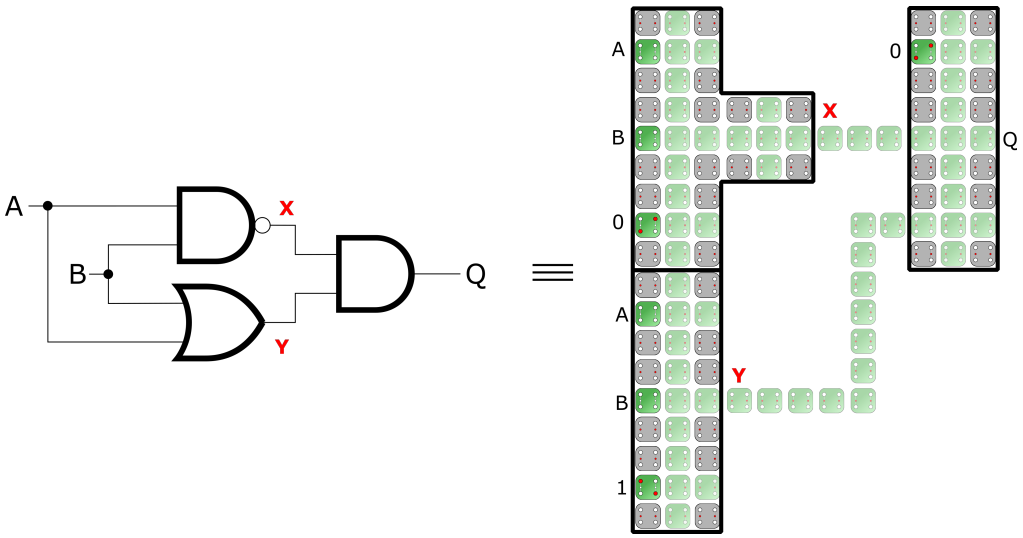


Figure 5.22: Translation of the XOR structure into RFCN cells

5.5 Reset islands

RFCN standard-cells are the building blocks used for more complex operations. More sophisticated operations with respect to inverter and majority voter will exploit multiple cells, therefore a bigger area. An interesting standpoint can be to look to a "zoomed out" view of cells. The standard-cells are placed in a regular scheme, and it can be noticed how always-on-reset containers (situated in the corners of standard-cells) generate *reset islands* (fig. 5.23.
Logically, the clock signals that drives the "reset islands" can be short-circuited and statically keep the containers on reset (fig. 5.24).

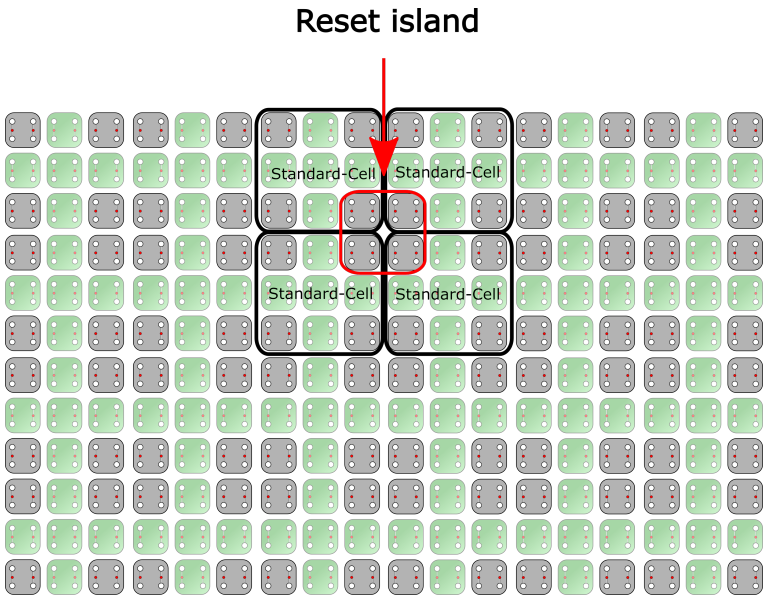


Figure 5.23: Nearby standard-cells generate reset islands

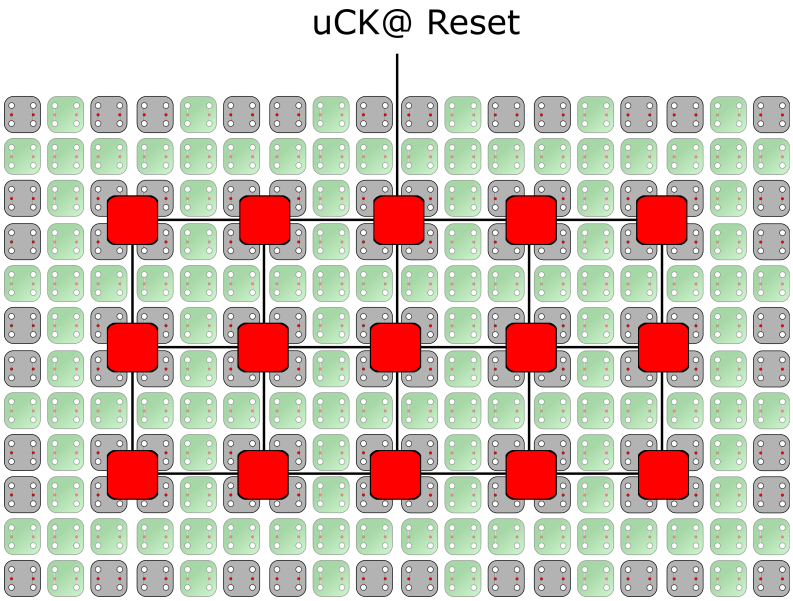


Figure 5.24: Reset islands can be short-circuited to a single clock signal which is kept at reset

5.6 Proposed design style: NAND logic

As previously mentioned, the synthesis and mapping consist only in choosing the proper electrical signals (u-clock and s-clock). The physical layer is a separated entity, hence it is totally independent from the synthesized network. This possibility allows for an indefinite amount of

design styles, thus in this work only one is proposed: NAND clusterization. In logic, functional completeness is the propety of a logic operator to be able to express all possible truth tables. One of the operators retaining this characteristic is the NAND gate [12]. This work proposes a NAND clusterization scheme as a possibility to implement all the primary capabilities of a VLSI circuit.

5.6.1 Clusterization scheme

According to the suggested NAND clusterization scheme, a delimited area is chosen, enclosing a certain number of containers. This *cluster* is "instructed" to perform specific tasks, including a NAND logic computation. Fig. 5.25 illustrates a recurrence of clusters.

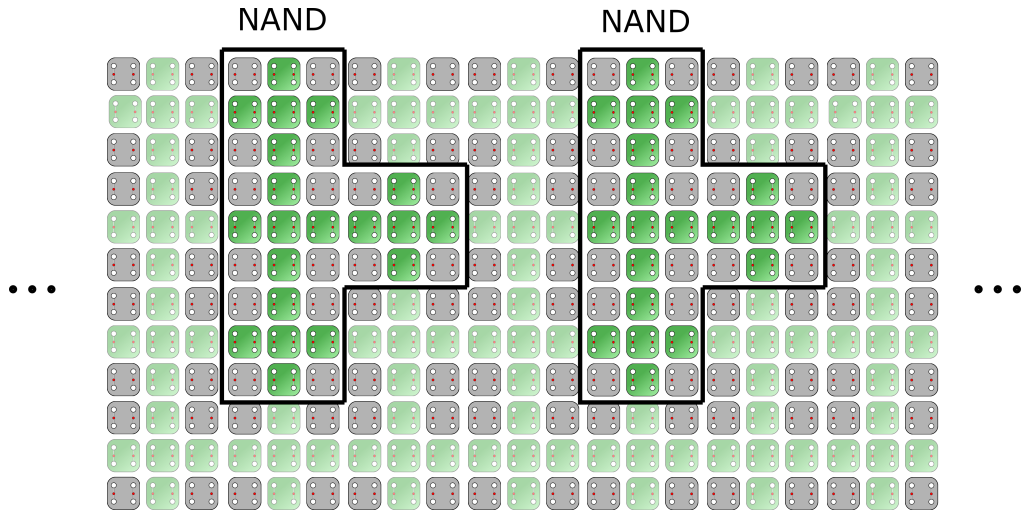


Figure 5.25: NAND gates repetition inside the Containers Common Layer

5.6.2 Why clusterization

This work believes that clusterization balances both the possibility to create custom logic routines, and the necessity to maintain a simple and repetitive clock scheme. The latter factor is crucial to avoid crowding during the clock wires fabrication: cells act in a correlated manner, otherwise, each cell would have to be "guided" independently from the others. The implementation of the clock scheme could be, then, very challenging.

5.7 Routing of data

A logic operation always requires two elements: the operator, and the operands. Indeed, to see the expected value on a cluster's output, the input values need to be the right ones. This section discusses how the data can travel between the clusters without needing additional wires.

5.7.1 Containers blockages

The structure presented up to this point leave no room for data routing, however, it is enough to dedicate a predefined number of containers that will not perform any logic operation, instead they will take care of propagating the logic state from a start to an end point. A portion of the area needs then to be reserved for data routing, and the suggested approach is to devote the containers surrounding the NAND gate for information propagation purposes. Fig. 5.26 displays a NAND operator surrounded by a free area.

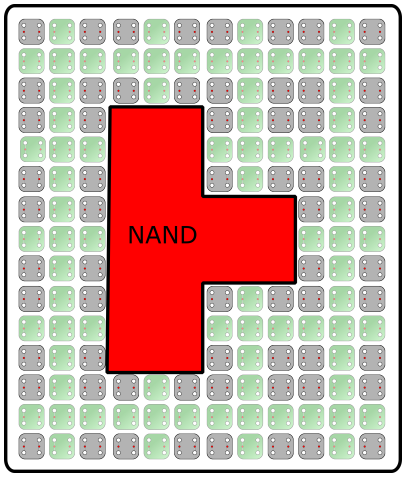


Figure 5.26: Blockages: around the operator, an area is reserved for non-logic operations. NAND cluster includes the blockage area

5.7.2 Memory islands

In addition to data propagation, can be convenient to retain specific containers and use them to hold precise data. As an example, a stored 0 can be propagated to the input of a majority voter and transform the former into an *AND* gate. To retain this extra containers, more area is occupied on the CCL.

5.7.3 Space and time multiplexing

Memory islands and information propagation occupy area on the Containers Common Layer (CCL). If a memory island is situated on the data propagation path, there is a *collision*. However, the flexibility of RFCN allow them to coexist on the same area by means of different methods. Two alternatives have been identified and presented: space and time multiplexing.

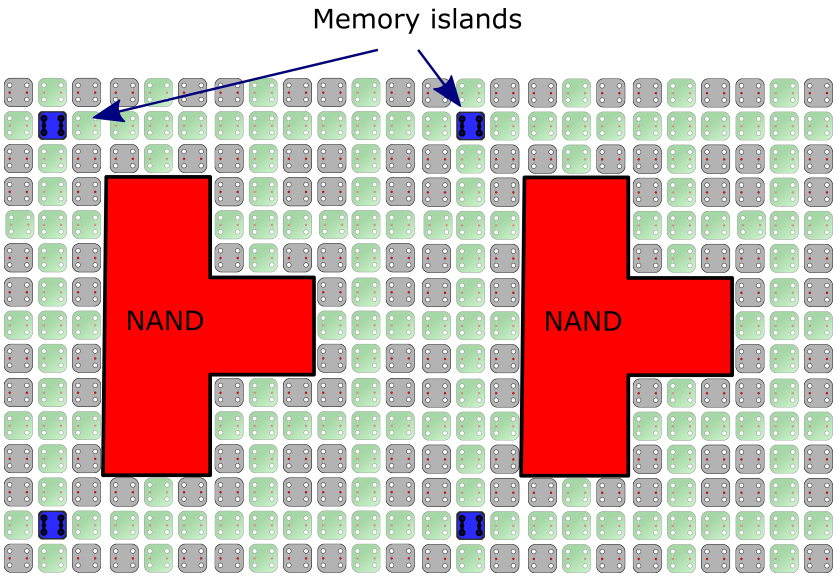


Figure 5.27: Memory islands

Space multiplexing

This technique extends the area of the blockages. This way data propagation paths and memory islands will never collide. However, the amount of area not exploited for logic operations increases. This means that less operations per second can performed by the same area.

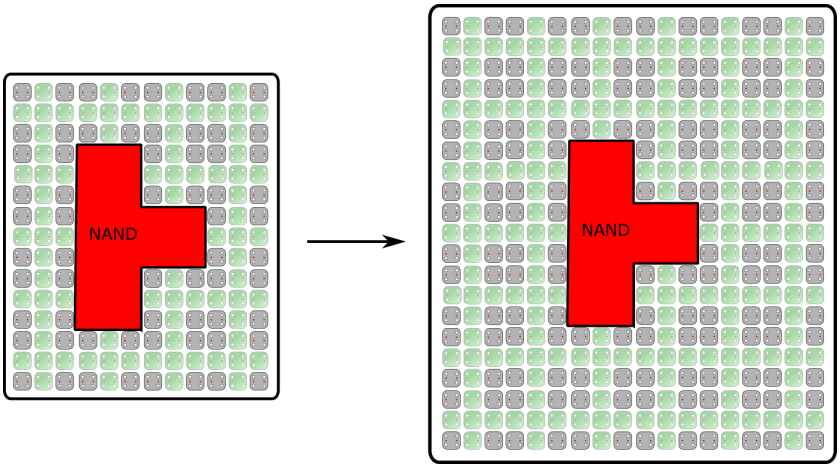


Figure 5.28: Extension of area blockage around NAND to perform space multiplexing

Time multiplexing

Time multiplexing can be exploited to avoid an excessive area utilization. It makes possible the propagation on data along a path that exhibits one or more memory islands. Provided to have a backup copy of the data stored in a nearby container, it is possible to temporarily wipe the data contained in the island and retrieve it back when no information has to travel in that zone. Fig. 5.29 shows the starting and endpoint of the bit to be propagated, while a time evolution is shown in fig. 5.30.

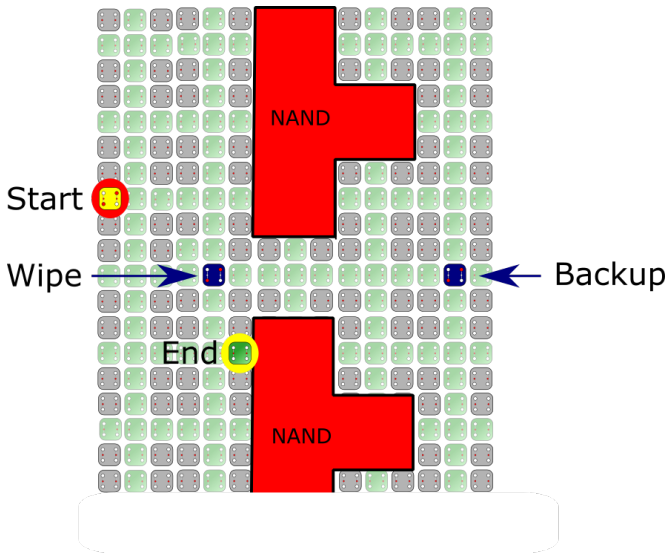


Figure 5.29: Time multiplexing: a bit has to travel from a *start* to an *end* point. A memory island is situated on a possible path, and a backup copy is nearby.

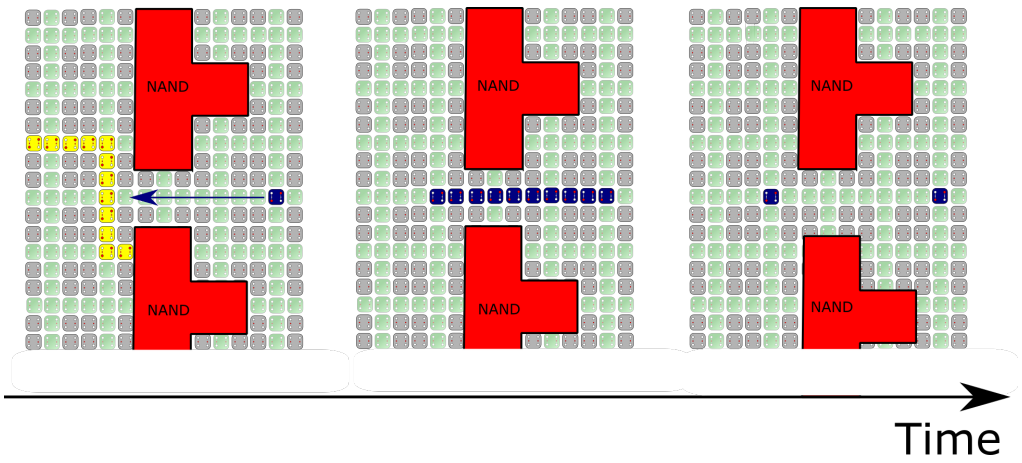


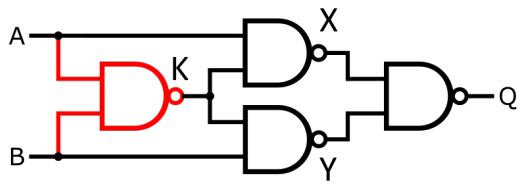
Figure 5.30: Evolution of the time multiplexing. The memory island is wiped, and when the data has travelled it is restored from the nearby backup copy

5.7.4 Simulation of operation and data routing

In order to clarify the behaviour of containers, it is performed a conceptual simulation of a XOR operation. The function mapped in RFCN is, however, the NAND-made version of the XOR, which is displayed in fig. 5.31. Four zones of containers are exploited for NAND operators. In order for the NAND to take place, the third input must be a logic *0*. The *0* value is stored inside the memory islands. In this simulations, memory islands share the same area of data propagation paths: a time multiplexing of routing resources is then necessary. The simulation presents four frames of the whole operation. In the first frame, shown in fig. 5.31, the inputs A and B are propagated to the input of the first NAND gate (NAND_1), and the 0's store in memory islands are provided to the same gate as third input. The value of the inputs, for the particuar example, is:

$$A = 1$$

$$B = 0$$



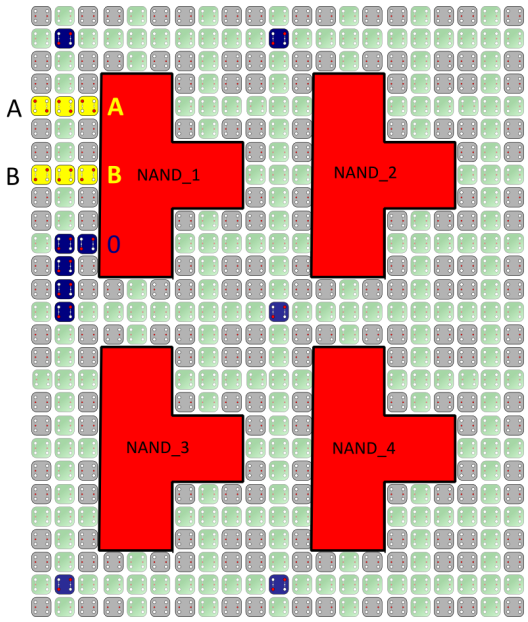


Figure 5.31: XOR - Step 1: input data enters the first cluster. A memory island provdes the third input to transform the majority voter into a NAND gate

In the second frame, the inputs arrive at the NAND_2 and NAND_3 gates, while the "third inputs" of the mentioned NAND gates are propagated from the nearby reset islands, as displayed in fig. 5.32.
In the third frame presented in fig. 5.33, the output of NAND_1 (node *K*) is propagated to NAND_2 and NAND_3 while the inputs A and B on the same gates remains memorized. According to the logic, the value of the node K is:

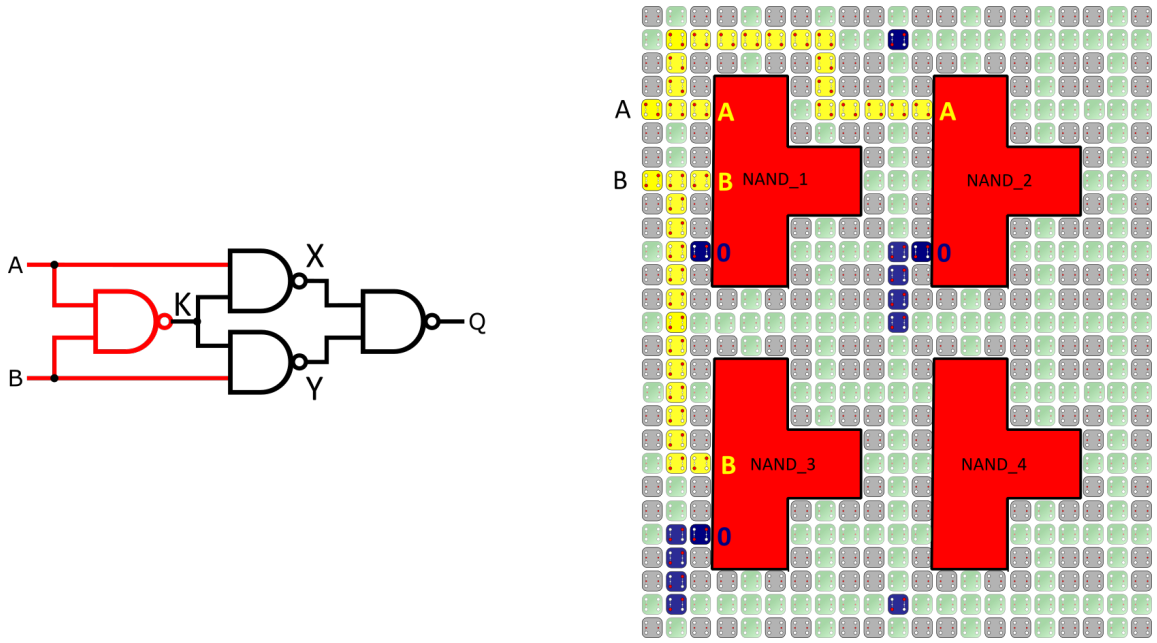


Figure 5.32: XOR - Step 2: while the first cluster performs the operation, binary data is propagated to *NAND_2* and *NAND_3*.

$$K = \overline{A \cdot B} = 1$$

Finally, the fourth and last frame shows in fig. 5.34 how the outputs of NAND_2 (node X and NAND_3 (node Y) is propagated to inputs of NAND_4, who releases the final output of the XOR operation (node Q). The values of X, Y and Q is calculated as:

$$X = \overline{A \cdot K} = 0$$

$$Y = \overline{B \cdot K} = 1$$

$$Q = \overline{X \cdot Y} = 1$$

Notice that the "tail" of the propagations are shown only for sake o simplicity, but in a real propagation those containers can be disabled after they perormed their task.

5.8 Hierarchical view

To have a clear picture of what RFCN looks like in a hierarchical perspective, figure 5.35 is introduced. The classic abstraction levels of VLSI circuits (from technology to register-transfer level) are mapped with the equivalent RFCN concept and the corresponding CMOS schemes. The basic "device" is represented by the charge container. In CMOS, the device is represented

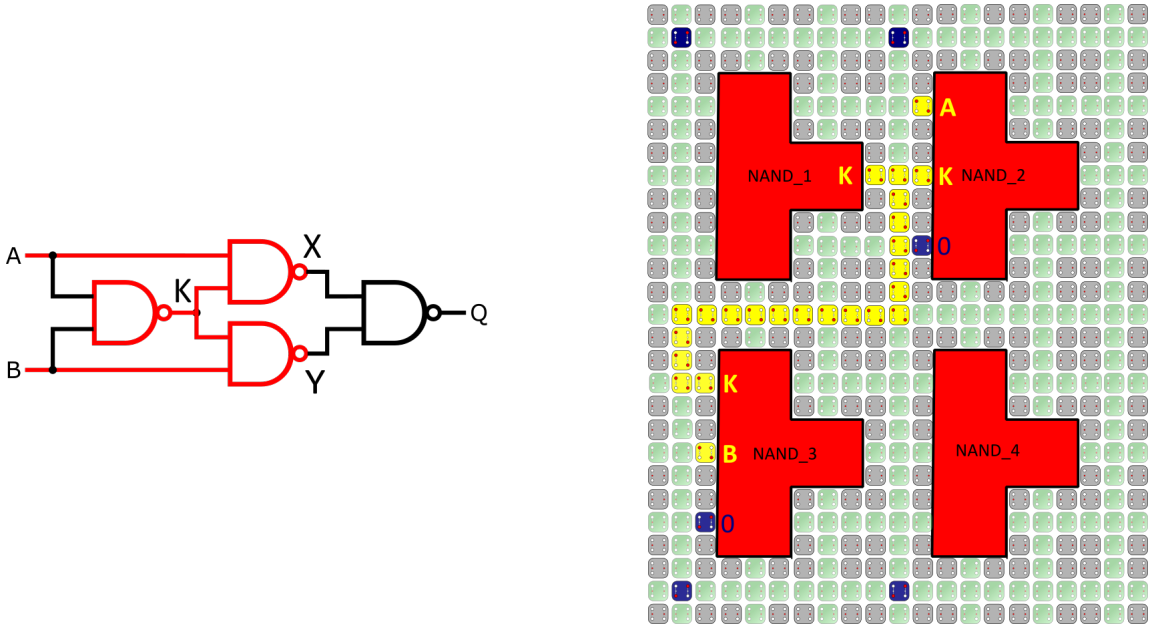


Figure 5.33: XOR - Step 3: A and B inputs are memorized to cluster's input area waiting for synchronization with results of the previous gate

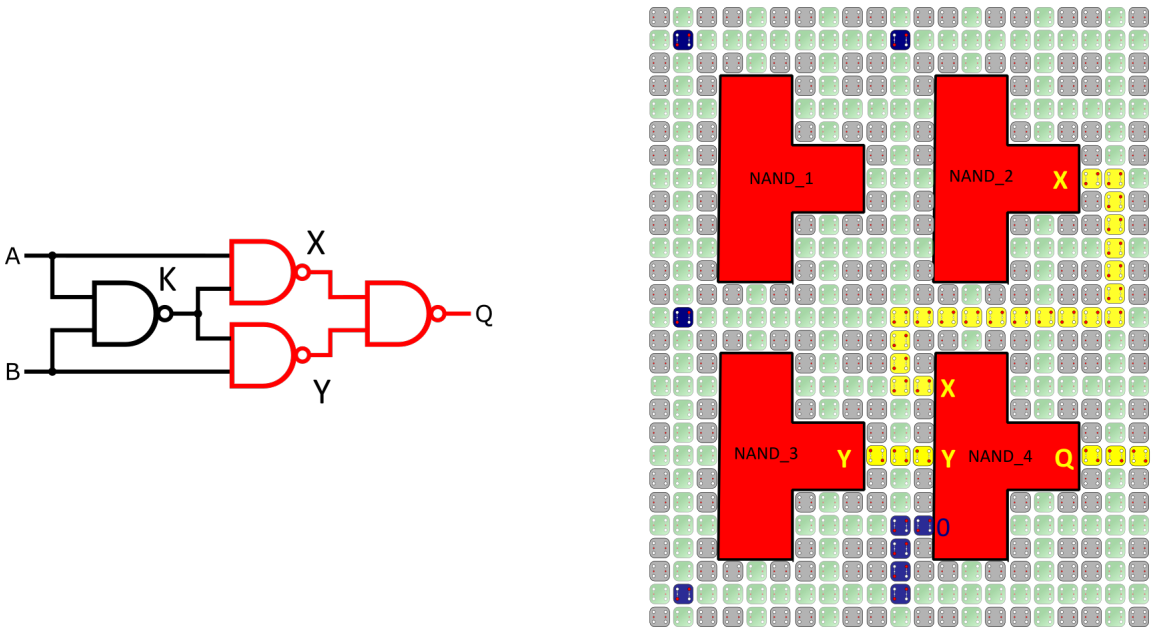


Figure 5.34: XOR - Step 4: The last NAND gate performs provides the result of the XOR operation

by the *transistor*. However, the charge container has intrinsic memorization properties: it is at the same time the equivalent of a *latch* or *flip-flop*. The logic gate is matched with the concept of *standard-cell*: the element able to perform basic logic functions (Inverter, AND,

OR, and Majority Voter). Since it is built by containers, which are also bit storage elements, it is associated with a clock edge-triggered element. The technology level, i.e. the physical implementation of the device, is discussed in chapter 6.

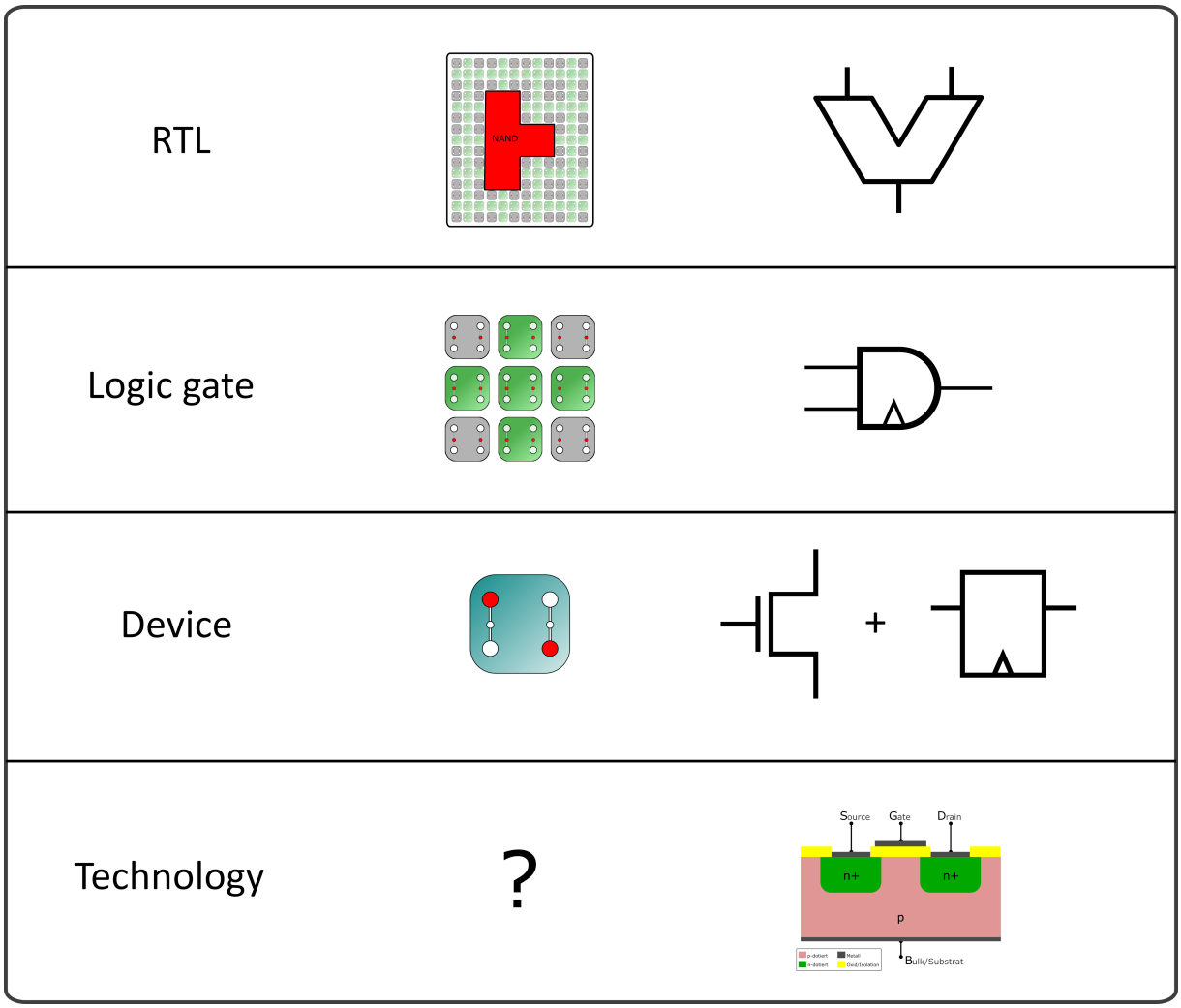


Figure 5.35: RFCN hierarchical view

5.9 Possibilities unlocked by RFCN

The design style proposed isn't intended to be exhaustive or to be the only solution. However, it presents some features that are currently harder to obtain on CMOS technology.

5.9.1 Reconfigurability

The key difference between RFCN and the other technologies mentioned in this work (CMOS, FCN) is how the logic gates are built. An inverter that is placed according to CMOS and FCN is physically shaped on a die ("written in the silicon"). An inverter operation in RFCN can instead be built at run-time, it is only necessary to change the timing of the electrical signals (uCK). In fact, the operation performed is determined by what micro-operations are stored in the sequencer, and provided to think to the sequencer as a memory, they can be replaced run-time. As a result, a NAND operator localized in a certain zone, can change into an OR operator (or any other) during the consequent SCK cycle.

5.9.2 In-memory computing

As result of the introduction of an *enable* signal (the clock), RFCN has intrinsic memorization properties. An example is the memory islands: they can be used to store any kind of information and access it at any time. It, therefore, a possibility to implement in-memory computing.

Chapter 6

Molecular implementation of the new RFCN

Reconfigurable FCN is a computing paradigm that introduces new possibilities, precluded with CMOS technology. Its physical implementation is suitable to any device capable of holding/releasing a charge by means of an enable command. Suitable technological implementations can possibly include Nanomagnets, Quantum Dots, and any future technology that can take advantage of RFCN scheme. This work, instead, describes a particular candidate for RFCN implementation: the molecular technology. All features introduced with Molecular FCN in Chapter 3 are inherited and expanded, according to the new RFCN paradigm.

6.1 Molecules advantage

In the FCN framework, it is believed that molecules can fit very well into the role of containers, due to their charge control properties. Moreover, a validating simulation scheme for FCN and Bisferrocene can be partially re-used to legitimize the molecular utilization in RFCN (see Chapter . Another reason is that, due to the possibility of placing molecules through SAM procedure, there is a good match with the concept of *Containers Common Layer (CCL)*.

6.2 Technology mapping

There are different scenarios to cover when talking about the technology mapping between RFCN and its molecular physical implementation. The first concept to translate is the idea of the charge container, using molecules, which is only partially described in section 3.1. A container has four charge dots, while each Bisferrocene molecule has two charge dots. Therefore the container is matched by two molecules. The clock field is applied by two electrodes: one electrode lying above the molecules, while the other coincides with the gold plate on which molecules are placed. Figure 6.1 illustrates the mapping from the container to the layout. However, the distance between two bisferrocene molecules is 2 nanometers [13]. While such a small

dimension goes towards an increase of *performance per area*, the size of a such electrode could cause two manufacturing issues:

- Clock wires congestion
- Electrodes size

6.2.1 Clock wires congestion

In order to apply the correct electric field, each electrode expects a clock wire (belonging to the micro-clock hierarchy). The high density of independent electrodes on the *SAM* area could cause the congestion of the micro-clock wires. In fact, since each electrode placed on a container will need a proper connection to somewhere else, the density of wires per unit area could saturate.

6.2.2 Electrodes manufacturing

The clock field that guides molecules state is handled by a metal electrode. The electrodes dimensions should, therefore, match the size of the single molecular container of $4\,\mathrm{nm}^2$, but the required accuracy for such a small electrode can result challenging during the manufacturing. However, section 6.3 explains how can be possible to enlarge the electrode and nevertheless have the same behavior of a single, bigger container called *molecular block*.

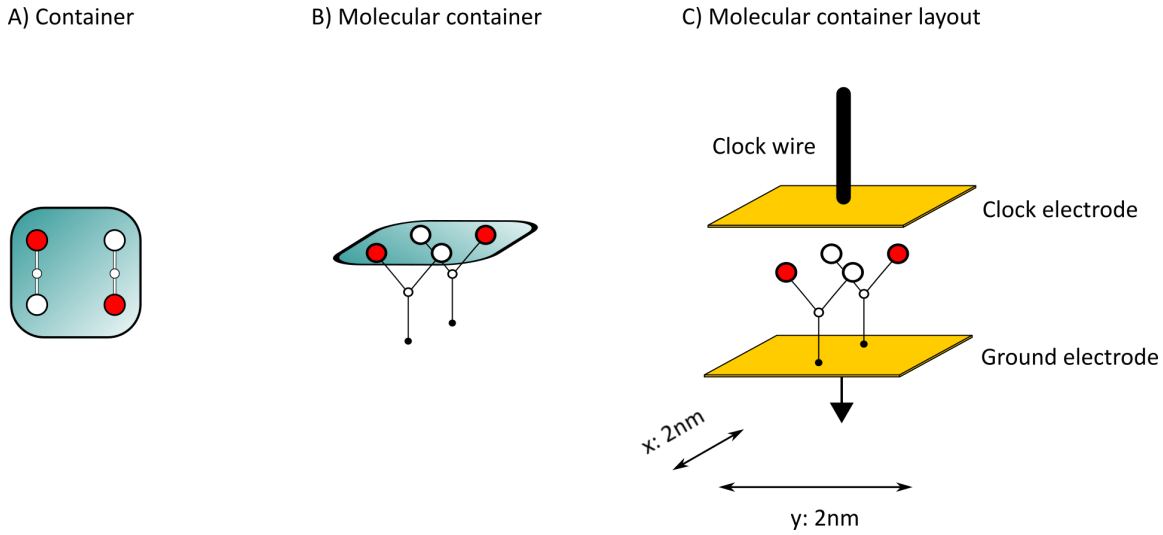


Figure 6.1: Possible layout of a molecular container made by two molecules

6.3 The molecular block

From an intuitive standpoint, and supposing they are subject to the same clock field, containers can be joint together and still act like a single container (fig. 6.2), according to their repulsion properties. Therefore, the single container (built by two molecules) can be combined with other containers to create a bigger, arbitrary, set of molecules. The prerequisite to make sure all the molecules act like a single group is to apply on them the same vertical electric field (micro-clock). The concept of a molecular container made by multiple entities, clocked by the same electrode, is named as *molecular block* (fig. 6.3). The boundaries of the molecular block are set by the dimension of the upper-lying electrode.

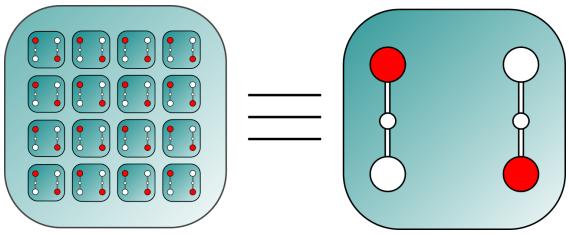


Figure 6.2: A set of containers that shares the same enable signal is equivalent to a single container

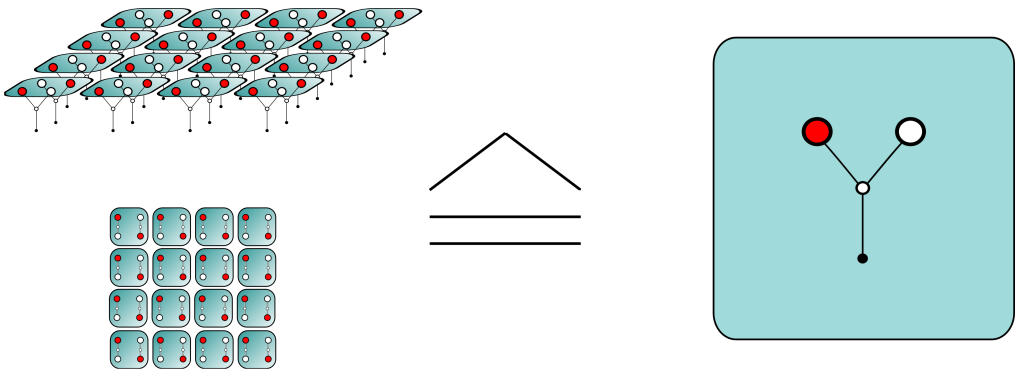


Figure 6.3: Symbol of a molecular *block*. A block can be built by an arbitrary number of molecules, provided to keep the same square ratio. A block behaves like a single container driven by a clock

6.3.1 Block dimensions

A reasonable constrain that can placed on the electrodes is regarding their shape: if they are squared, a standard pattern can be more easily realized. The number of molecules that build the block should then be such to keep a square aspect ratio. Because the distance between two molecules in the y direction is doubled with respect to gap on the x direction [3], the number of molecules lying on x should be double. If N is the amount of molecules that lie on the x

direction, the number of molecules on the *y direction* shall be, then, $N/2$. The size of the block, as the size of the upper-lying electrode, is a quantity that can affect many characteristics subject of analisys in this work. Hence, N is kept as parameter.

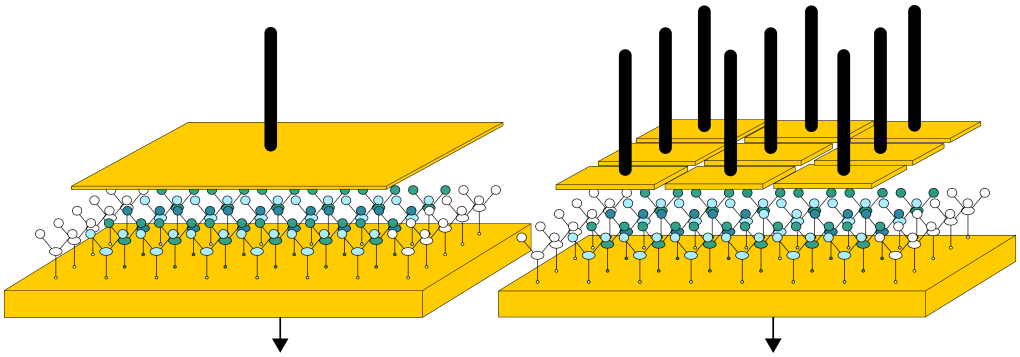


Figure 6.4: Layout of a molecular block (left). Layout of nine smaller molecular blocks (right). Depending on the size of the electrode, the number of molecules in a block changes accordingly

6.3.2 The molecular micro-clock

As mentioned in Chapter 3, the clock field is applied to molecules through two electrodes: one lying on the top, and the ground plate. When a positive signal is applied, molecules push their charge upwards, otherwise downwards, as shown in fig. 6.5. The temporization of that field estabilishes the role of the molecular block inside the standard-cell. of voltages applied determines the enalbe/disable of the clock signal.

6.3.3 Molecular standard-cell

In Chapter 5 the concept of RFCN standard-cell is introduced. In Molecular RFCN the standard cell keeps the same organization, with an only exception: it is not made by nine containers but by nine molecular blocks (each one with N molecules). The representation for the molecular standard-cell is shown in fig. 6.6.

6.4 Electrodes sharing

During the building of the Molecular RFCN technology, several ideas have been figured out on how to reduce the needed number of indipendent electrodes. Even if RFCN can work properly with the scheme described until this section, reducing the number of independent electrodes could favor an easier management of the timing signals. A possibility for this purpose is to try to share the electrodes belonging to the same row and to the same column. The outcome was a kind of *battleship* game, in which the goal was to exploit electric field interference between

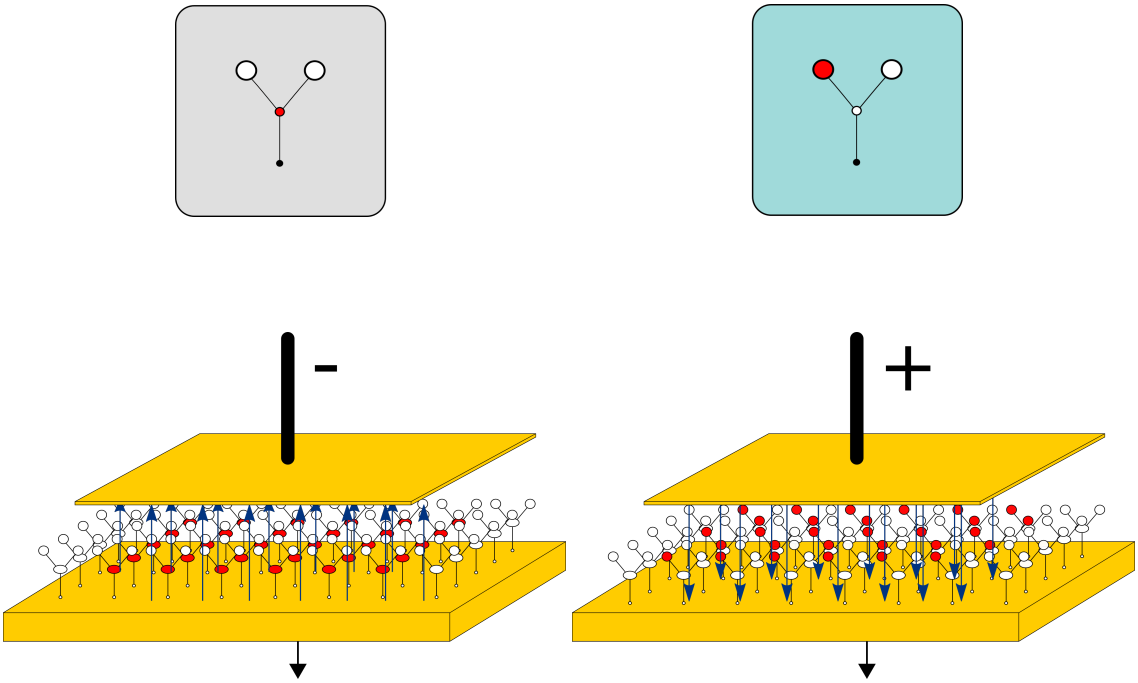


Figure 6.5: Negative charge localization in a molecular block. Reset fields pushes charge in the third dot (left). Hold field directs the charge in the upper dots (right). The opposite scenario applies in case of molecules having positive ionization.

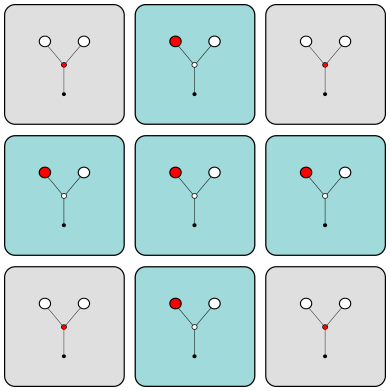


Figure 6.6: Standard cell concept in molecular RFCN: containers (2x2 molecules) are replaced by blocks (arbitrary molecules number)

rows and columns, and its scheme is shown in fig. 6.7. Note that this electrode sharing model does not involve the ideas presented previously (sequencer, microclock).
Unfortunately, even if this scheme could open to a simplification of the sequencer control network, it can not be applied in Molecular RFCN. When a molecule switches to a different logic state, it must pass through a reset state, otherwise, the nearby molecules would prevent it to change state. The reset state acts like a *cleaning* of the containers memory before the next hold

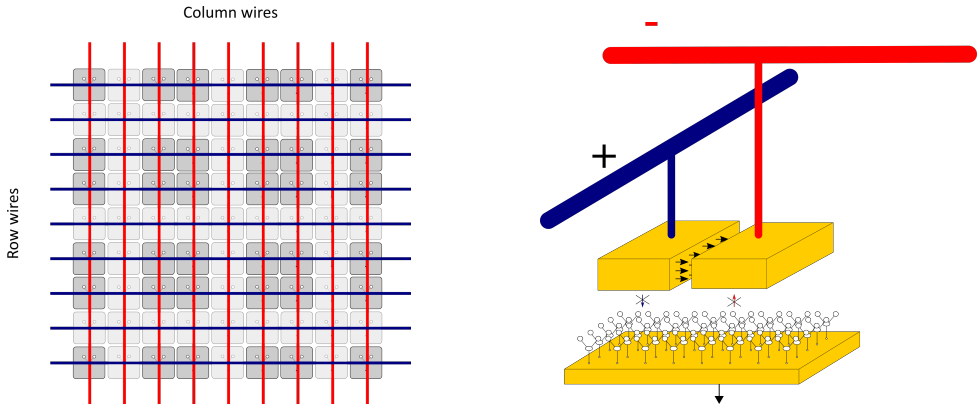


Figure 6.7: Row and columns electrodes shared model (left). Single block electrode layout according to the shared model (right). The interference between positive and negative voltage prevents the electrodes from exercising a force on the molecules, allowing the (row, column) pair to enable only blocks that don't experience interference.

state. Using a shared model, it was not possible to figure out a strategy to guarantee the needed degrees of freedoms required to mimick the turning-on sequence of the inverter and majority voter.

6.5 Manufacturing

Molecules can match the role of charge containers [2], and can allow performing all the operations needed to build complex logic. This section presents a potential manufacturing procedure, and the possible issues that it might present.

6.5.1 SAM

A self-assembled monolayer is a set of organic molecules that is formed *spontaneously* on a surface by adsorption [14]. In the case of Bisferrocene, the surface would be a gold plate. Molecules in a SAM are organized in an orderer domain [14], which makes the process very suitable for RFCN. Only after molecules are created on the gold layer, upper electrodes can be built.

6.5.2 Possible issues

Compared to FiNFET process, the SAM manufacturing steps do not require as many process iterations. However, process variability plays a role also in this technology. The non-ideality that can take place are, between the many [15]:

- Grain defects

- Molecules orientation
- Electrodes shape

Grain defects

The gold surface on which molecules will be deposited could experience roughness [15]. Its planar characteristics can be perturbed by the presence of gold grain boundaries [15]. Grains can have different shapes and they can lie at different heights. This vertical misalignment is transferred to the above molecules: dots belonging to the first molecule can have a further distance than expected from dots of the nearby molecule. However, only above a certain misalignment threshold this phenomenon can cause issues.

Molecules orientation

The simulations performed in chapter 7 rely on a coherent orientation of molecules. Nevertheless, if a molecule is tilted with respect to its expected angle, it could modify the real distance between two charges.

Molecules alignment

Another variability that could negatively affect the propagation is a horizontal dis-alignment of molecules [15]. It is not known which position molecules will have once deposited on the gold surface [15], hence this issue needs a Bisferrocene SAM prototype to be validated.

Chapter 7

Molecular RFCN simulations

This chapter illustrates through MATLAB® simulations how molecular blocks behave when subject to an electric field, and how they act when sequentially activated to perform logic operations. The simulations take advantage of the *Molecular Simulator Quantum-Dot Cellular Automata Torino (MoSQuiTo)* and *Self Consistent Electrostatic Potential Algorithm (SCERPA)*, developed at Politecnico di Torino [16] [3].

7.1 Simulation flow

In order for the interaction between molecules to be correctly simulated, a three-stages algorithm is applied [3].

7.1.1 Stage I: AB-initio characterization

In the first stage, the *Bisferrocene* molecule is characterized through *AB-initio* (atomistic) simulations [3]. The molecule is described through *Z-matrix*. Z-Matrix encodes the information by providing the relative position among atoms. The Z-Matrix that describes a portion of the molecule is shown in table 7.1.

Atom symbol	**Atom reference #**	**Distance [Å]**	**#**	**Angle [°]**	**#**	**Angle [°]**
C	1	1.424				
C	2	1.424	1	108		
C	3	1.424	2	108	1	0
C	4	1.424	3	108	2	0

Table 7.1: Z matrix describing a Ferrocene group

The resulting molecular structure is shown in fig. 7.1.

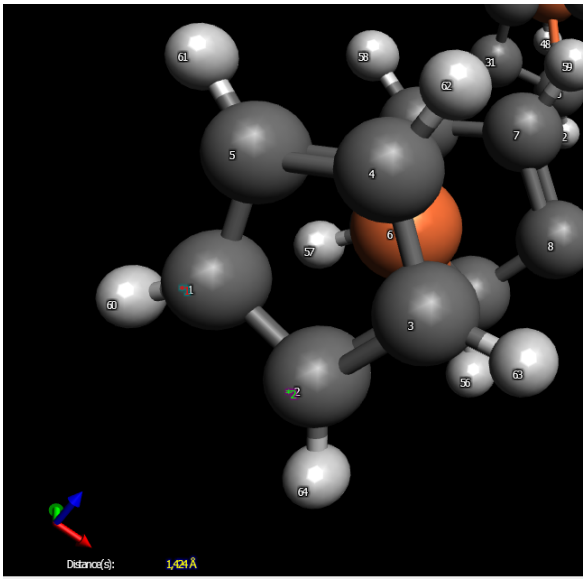


Figure 7.1: Atoms arrangement of a single *ferrocene* group inside the molecule

During the atomistic simulation, the molecule under test is subject to biasing conditions: an electric field with different values is given as "input" to the simulation tool [3]. The input field is created by two point charges, whose position mimics the charge distribution of a nearby molecule, called *driver*. The output electric field generated by the molecule under test is used as simulation data and given as input to the II step of the flow.

7.1.2 Stage II: Aggregated charge model and transcharacteristics

The charge distribution in the molecule under test is strongly affected by the input field of the *driver* molecule. However, since each electron might only move slightly, it is the overall effect of the electron mass that determines a change in the charge distribution [17]. Aggregate charge model performs a mean operation among the molecular charge. As a result, red-ox centers are modelled like a point charge having, as value, the average of the surrounding charge [17]. Fig. 7.2 displays the charge distribution and the aggregated charge model of a molecule in a "hold" configuration.
Moreover, the results of Stage I allow to build trans-characteristics: for each input voltage value, the equivalent output voltage value of the molecule is calculated. The trans-characteristic graph allows the definition of a *gain*: it is calculated as the ratio between the equivalent voltage that the m.u.t (molecule under test) supplies to the next molecule and the voltage that the m.u.t is receiving from the previous. It depends mainly on two factors. The first is the molecular distance: the higher, the lower will be the intensity of the field arriving at the m.u.t, and the lower the m.u.t. will be able to forward. The second parameter is the presence of the vertical clock field: when it is active, enhances the *gain* of the molecule. Figure 7.3 shows the parametric

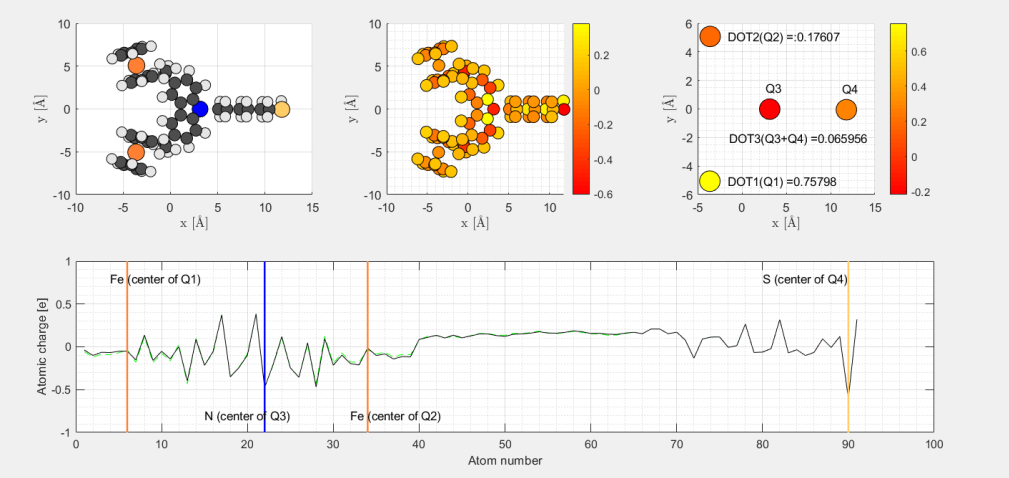


Figure 7.2: Molecular structure (upper left). Charge distribution around the single atoms (upper center). Aggregated charge value (upper right).

trans-characteristic (having the distance as a parameter), in presence and absence of the clock field.

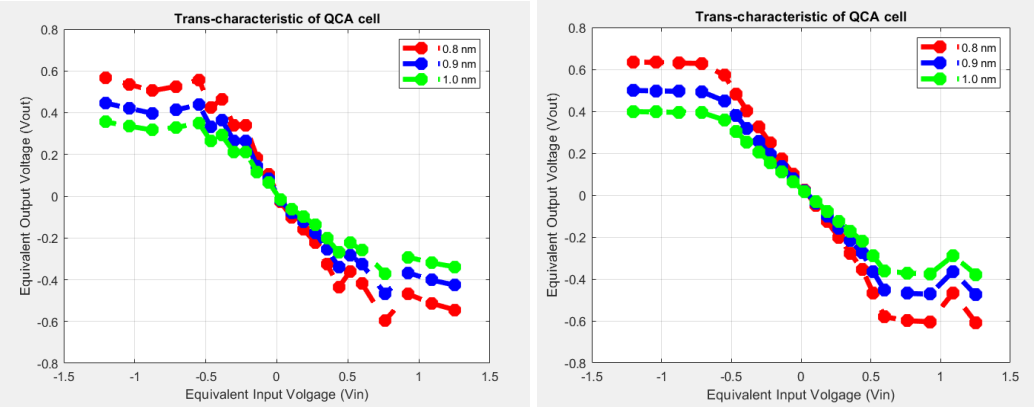


Figure 7.3: Effect of molecular distance and clock field on the trans-characteristic. The graph on the right shows how a clock field enhances the output voltage values of the molecule.

Considering as input voltage a value equivalent to the maximum output voltage of the molecule ($V_{IN} = 0.6V$), the resulting gain can be seen in table 7.2. Compared to configuration without clock applied, the values of gain are higher: the result explains the advantage of using a clock network.

Distance [nm]	Gain with clock +2 V	Gain without clock
1	0.67	0.58
0.9	0.821	0.731
0.8	1.046	0.93

Table 7.2: Equivalent voltage gain in absence and presence of a clock field

7.1.3 Stage III: Self-consistent algorithm

When subject to an electric field, molecules apply mutual force to each other. The electric field that propagates in a chain of molecules experiences a feedback effect on all of them, meaning that the true electric field can be only known when all the electric fields are known [16]. The III stage of the simulation algorithm consists of evaluating iteratively, using a self-consistent algorithm (SCERPA), the proper charge and field values among a whole set of molecules until a stable configuration is reached.

7.2 Simulation structure

This section explains how molecular circuits are described in MATLAB: this includes the pattern of the molecules, the clock zones, and the simulation parameters such as clock field strength and molecular distance.

7.2.1 Molecular standard cell structure

As a first step, the structure of the molecular circuit has to be described. The code in 7.2.1 illustrates the details of the circuit matrix. The text "Dr1" and "Dr2" corresponds to the driver molecules that input the standard cell. In this sample standard-cell, a single block is made by 4x2 molecules ($N = 4$), which means 4 molecules on the x axe and 2 on the y axe. A molecule is placed when a number different from 0 is set into the matrix. The value of the number indicates the clock signal to which molecules obey, which is generated by the above electrode. The corresponding molecular circuit is displayed in 7.4.

```
%% Standard cell structure

%N=4 (Size 4x2)

circuit.structure = {
        0  0    '1' '1' '1' '1' '2' '2' '2' '2' '3' '3' '3' '3'
        0  0    '1' '1' '1' '1' '2' '2' '2' '2' '3' '3' '3' '3'
    'Dr1' 'Dr2' '4' '4' '4' '4' '5' '5' '5' '5' '6' '6' '6' '6'
    'Dr1' 'Dr2' '4' '4' '4' '4' '5' '5' '5' '5' '6' '6' '6' '6'
        0  0    '7' '7' '7' '7' '8' '8' '8' '8' '9' '9' '9' '9'
        0  0    '7' '7' '7' '7' '8' '8' '8' '8' '9' '9' '9' '9'
    };
```

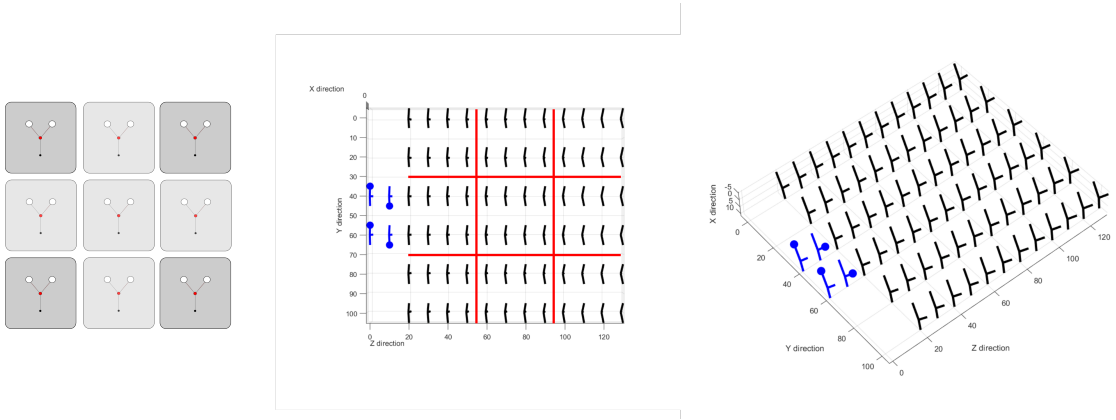


Figure 7.4: Molecular standard cell (left). Side view and top view of the molecules' simulation geometry (middle, right).

7.2.2 Molecular distance

The distance between two consecutive molecules is an important parameter, useful to predict the behaviour of molecules in case of process variations. It described in MATLAB thorught the code lines in 7.2.2.

```
13          %Molecular distances
14          circuit.dist_z = 10; %Angstron
15          circuit.dist_y = 20; %Angstron
```

7.2.3 Clock evolution

After that clock zones have been identified and described, the evolution of the micro-clock signals over time has to be reported. Figure 7.5 shows the timing of the clock signals that allow to build an inverter, while 7.2.3 shows their description in MATLAB: each column represents a micro-operation in a given time stamp.

```
16          circuit.stack_phase(1,:) = [rst rst rst rst rst rst rst rst];
17          circuit.stack_phase(2,:) = [rst rst rst rst hld hld hld rst];
18          circuit.stack_phase(3,:) = [rst rst rst rst rst rst rst rst];
19          circuit.stack_phase(4,:) = [rst hld hld rst rst rst rst rst];
20          circuit.stack_phase(5,:) = [rst rst hld hld hld rst rst rst];
21          circuit.stack_phase(6,:) = [rst rst rst rst rst rst hld hld];
22          circuit.stack_phase(7,:) = [rst rst rst rst rst rst rst rst];
23          circuit.stack_phase(8,:) = [rst rst rst rst hld hld hld rst];
24          circuit.stack_phase(9,:) = [rst rst rst rst rst rst rst rst];
```

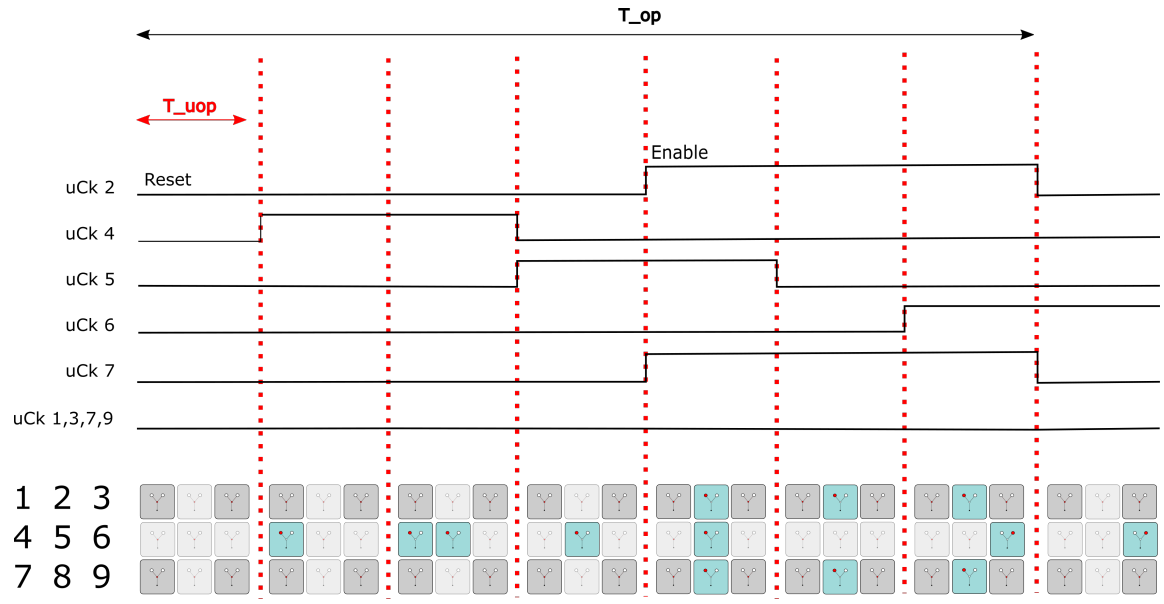


Figure 7.5: Evolution of the micro-clock signals and the corresponding standard-cell evolution during an inverter operation.

7.2.4 Drivers

The evolution of driver's configuration over time is descrbied in 7.2.4. For the inverter example, it is kept constant during the operation.

```
25          circuit.Values_Dr = {
26          'Dr1' -4.5 -4.5 -4.5 -4.5 -4.5 -4.5 -4.5 -4.5 'end'
27          'Dr2' +4.5 +4.5 +4.5 +4.5 +4.5 +4.5 +4.5 +4.5 'end'
28          };
```

7.3 Logic operations simulation

The simulation flow and structure illustrated in the previous sections is here used to validate the use of molecules to perform RFCN operations. The size of the molecular block is set to

$$N = 6$$

The figures that show the simulation progression are include the molecular circuit (top, side view) and the high-level description of the blocks turning-on sequence. The logic operations simulated are:

- Inverter

- Majority voter
- NAND

7.3.1 Inverter

The basic inverter operation can be performed using only one RFCN standard-cell, which is made by 9 molecular blocks. The images from fig. 7.6 to fig. 7.12 represent all the evolution of the cell while performing an inversion of the input data. Four schemes are shown for each picture, each one providing a different information:

- A): high level blocks representation and their progression
- B): binary value encoded by the molecular block, according to molecules charge distribution
- C): charge localization in the molecular circuit (top view)
- D): charge localization in the molecular circuit (side view)

The input value is

$$D_{in} = 1$$

that corresponds to the following molecular dots configuration:

$$\begin{array}{ll} Filled & Unfilled \\ Unfilled & Filled \end{array}$$

The expected output value is

$$D_{out} = 0$$

that corresponds to the following molecular dots configuration:

$$\begin{array}{ll} Unfilled & Filled \\ Filled & Unfilled \end{array}$$

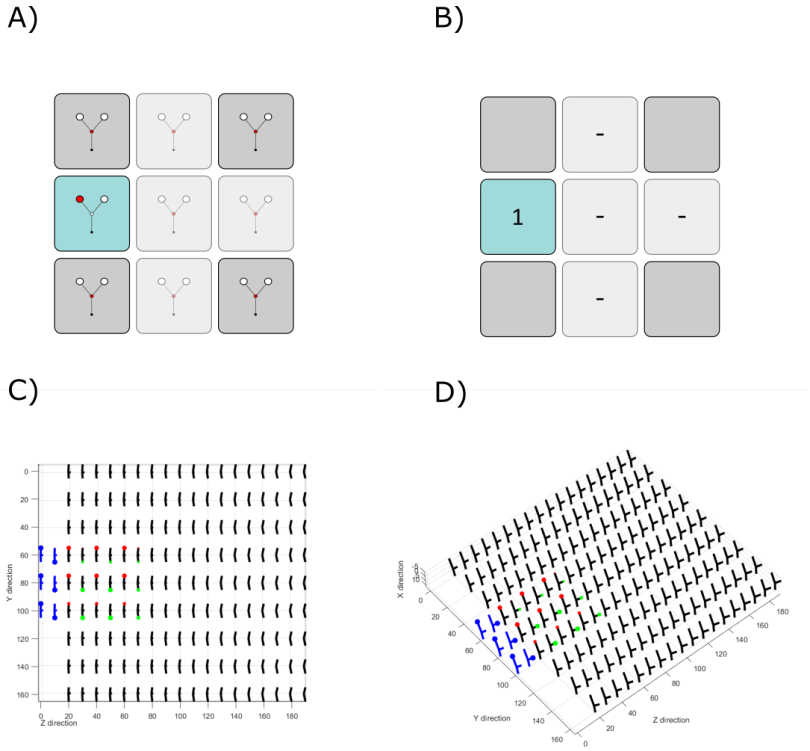


Figure 7.6: Simulation progression - Step 1 (Inverter)

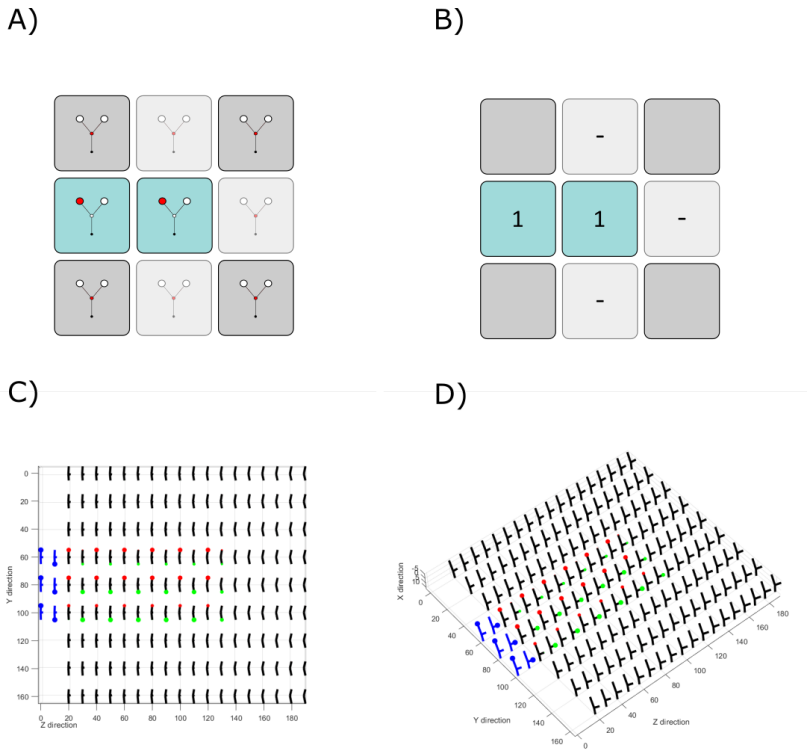


Figure 7.7: Simulation progression - Step 2 (Inverter)

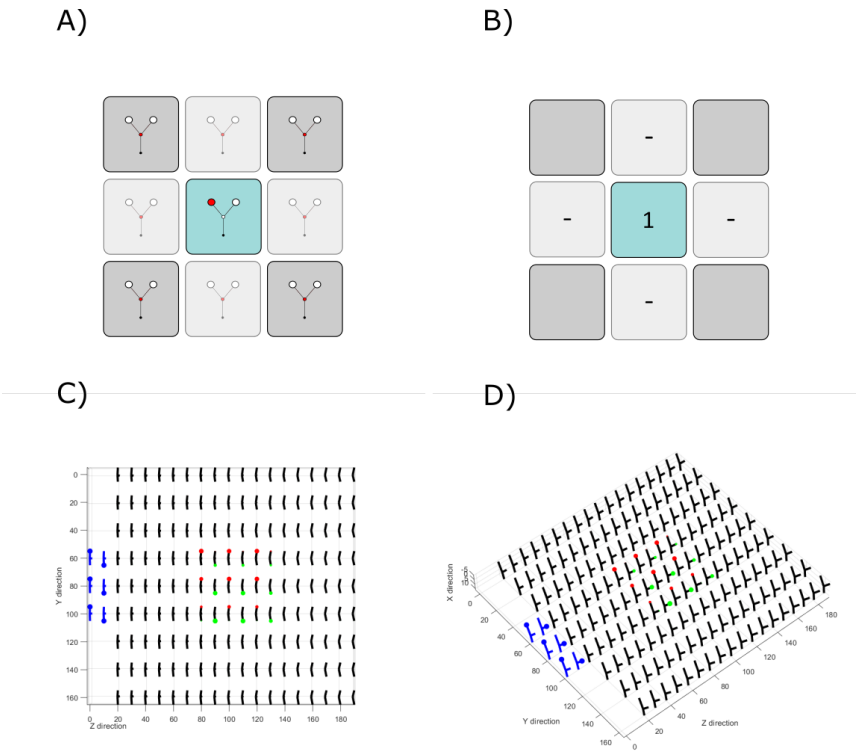


Figure 7.8: Simulation progression - Step 3 (Inverter)

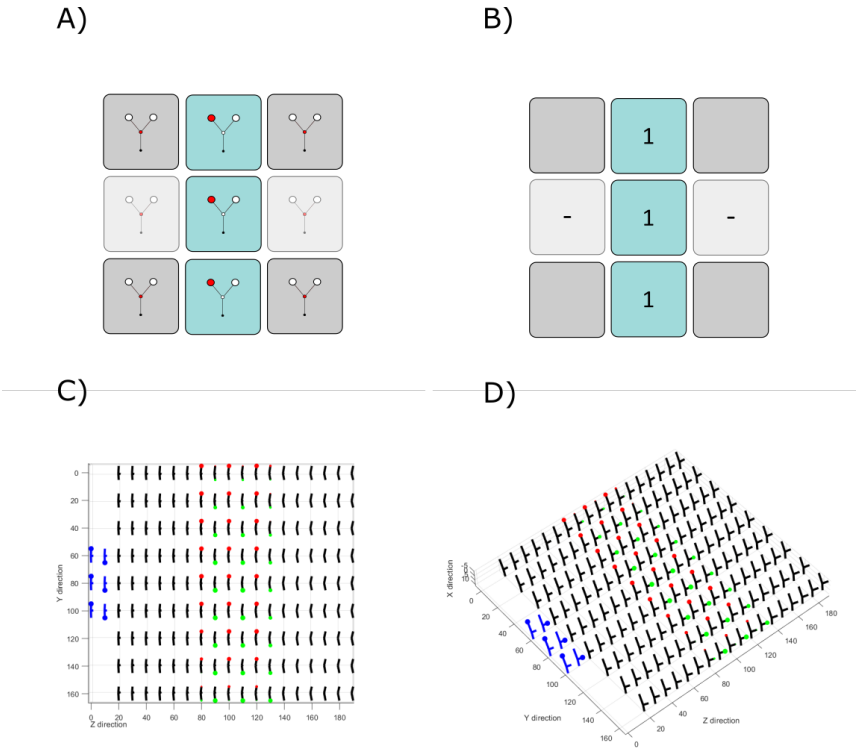


Figure 7.9: Simulation progression - Step 4 (Inverter)

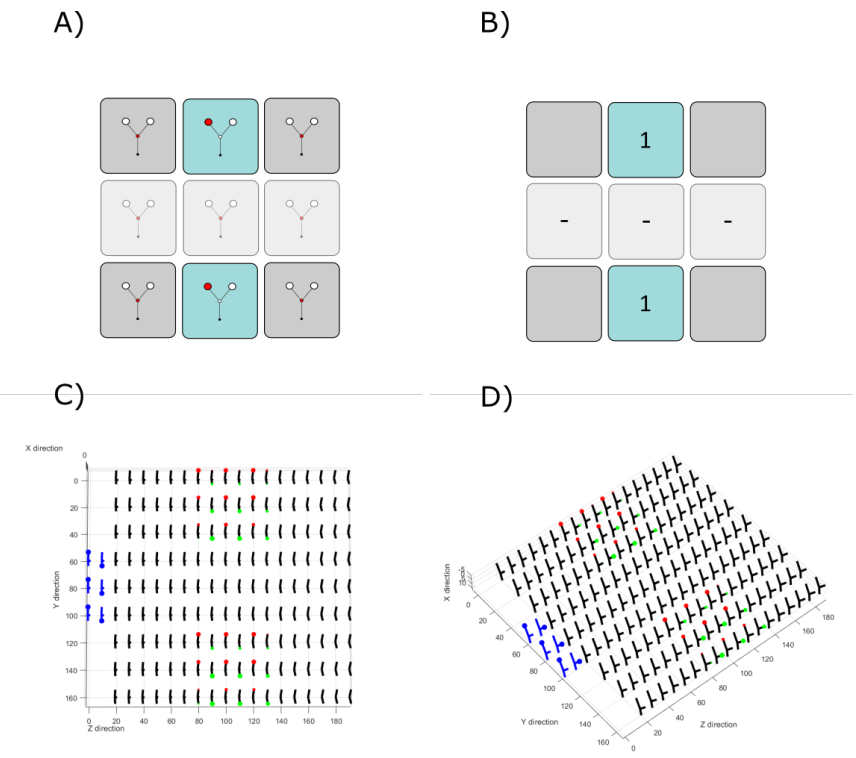


Figure 7.10: Simulation progression - Step 5 (Inverter)

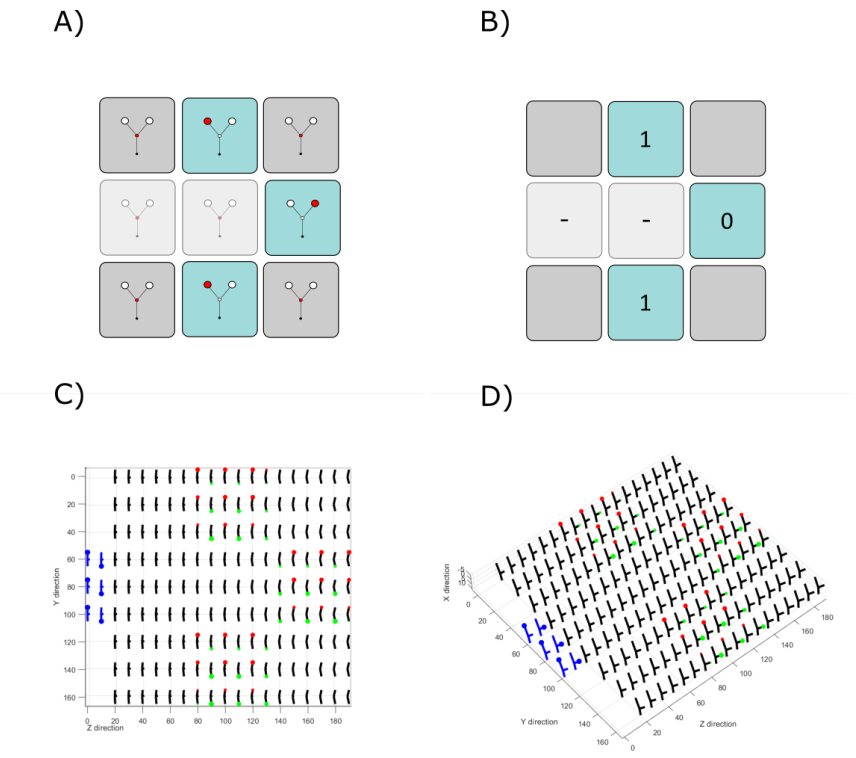


Figure 7.11: Simulation progression - Step 6 (Inverter)

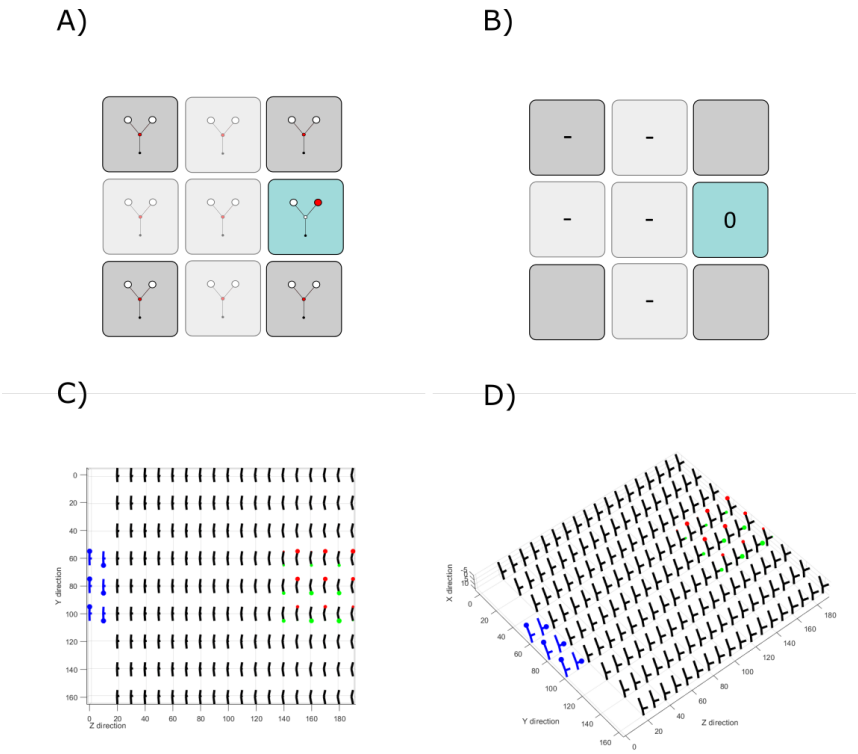


Figure 7.12: Simulation progression - Step 7 (Inverter)

7.3.2 Majority voter

The majority voter operation can also be performed using one standard-cell. The images from fig. 7.13 to fig. 7.17 represent all the evolution of the cell while performing the majority operation of the input data. The four images represent the same information. The inputs are D1 (top input), D2 (left input) and D3 (bottom input). The input values are

$$D1 = 0$$

$$D2 = 1$$

$$D3 = 0$$

The expected output is therefore

$$D_{out} = 0$$

that corresponds to majority of inputs (D1 and D3). The chosen logic value associated to the dots charges is kept the same as in 7.3.1

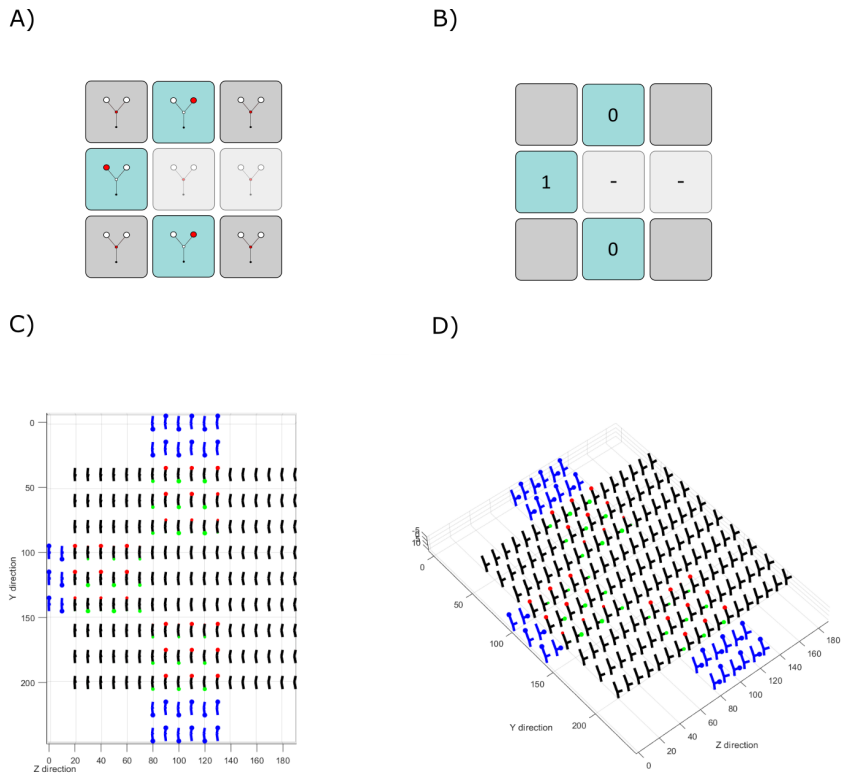


Figure 7.13: Simulation progression - Step 1 (Majority voter)

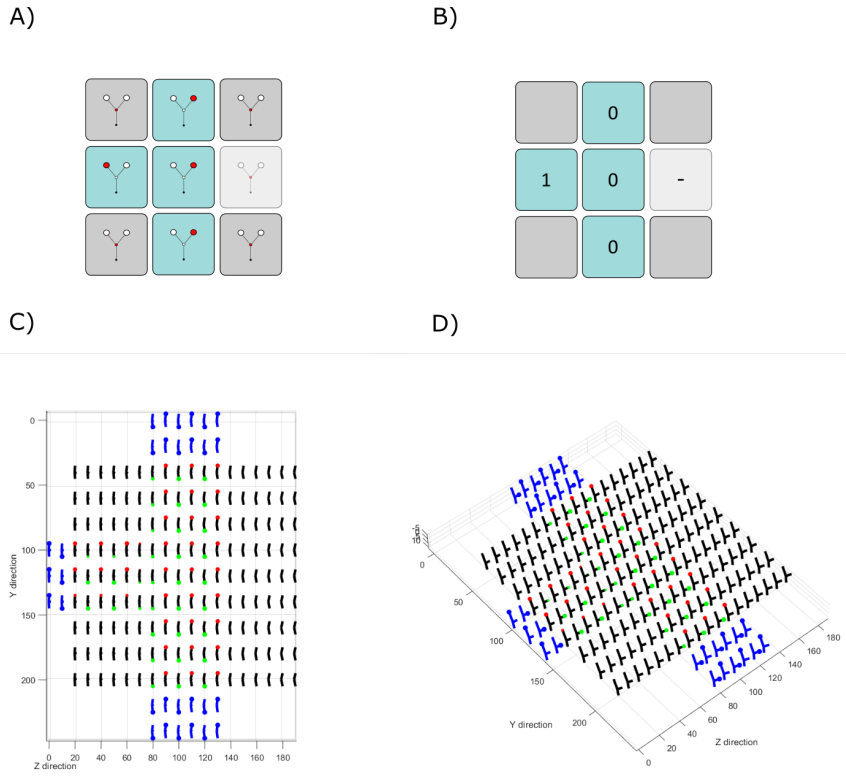


Figure 7.14: Simulation progression - Step 2 (Majority voter)

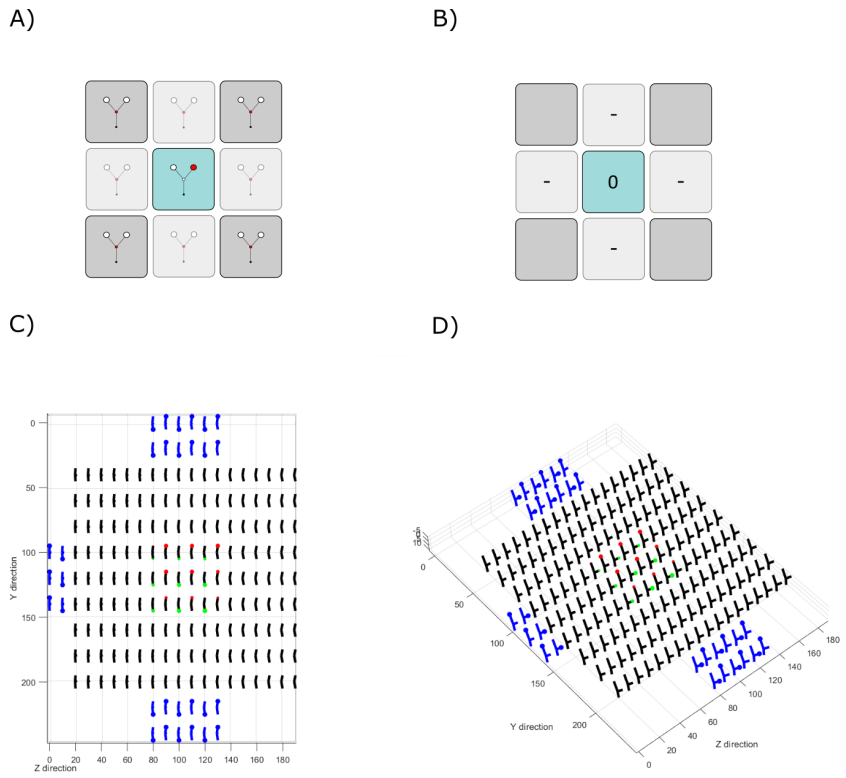


Figure 7.15: Simulation progression - Step 3 (Majority voter)

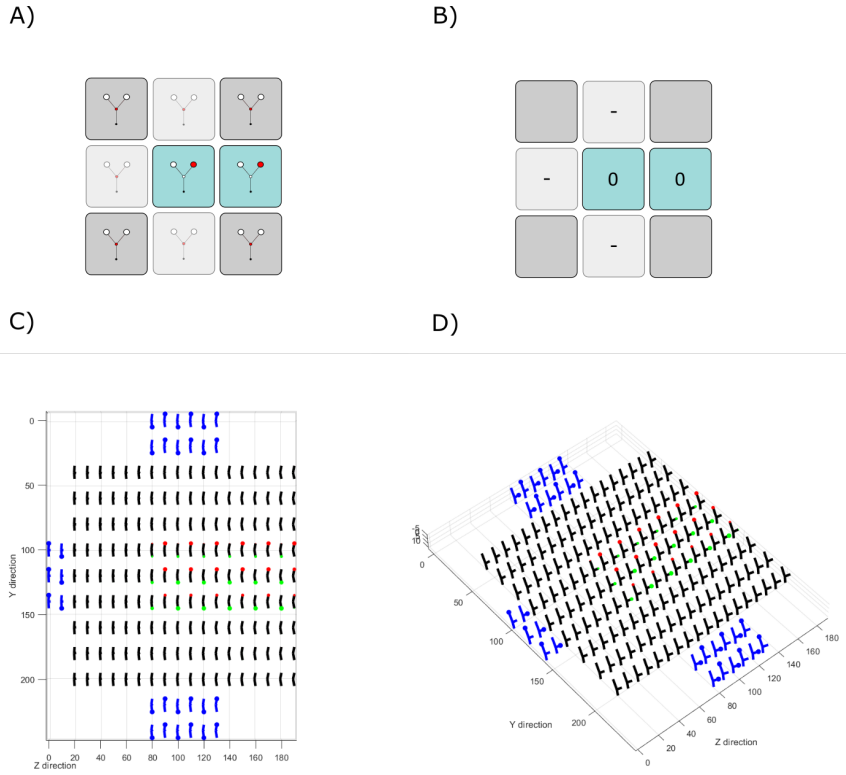


Figure 7.16: Simulation progression - Step 4 (Majority voter)

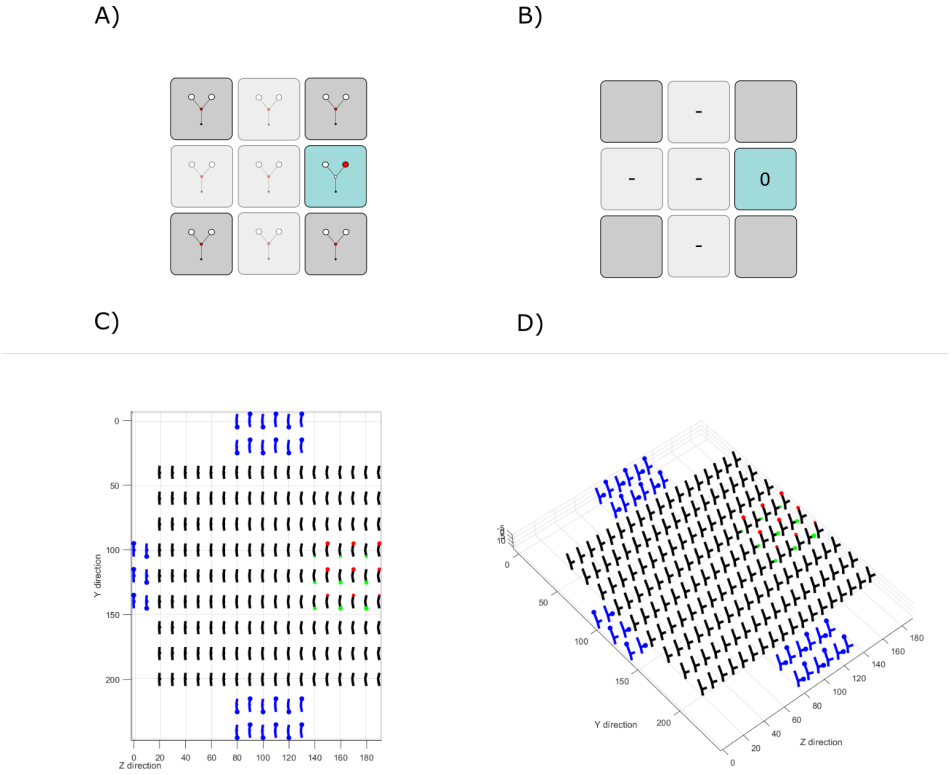


Figure 7.17: Simulation progression - Step 5 (Majority voter)

7.3.3 NAND

The two previous simulations basic operations performed within a standard cell, therefore they aim at validating the correct interaction between molecular blocks. In order to extend the validity of the model, another simulation is performed, aiming instead at legitimize the correct interaction between *standard cells*. For this purpose, a more complex logic gate is analyzed: NAND. It makes use of 6 standard cells, of which only 4 are used for the operations. The operations requires 12 steps, and the simulation progression is shown from fig. 7.18 to fig. 7.29. In order for the NAND to work correctly, the third input on the majority voter is a binary 0. The molecular circuit receives the 0 in the top-left corner. The inputs are A (middle input) and B (bottom input), and their value is

$$A = 0$$

$$B = 1$$

The expected output is therefore

$$Q = \overline{A \cdot B} = 1$$

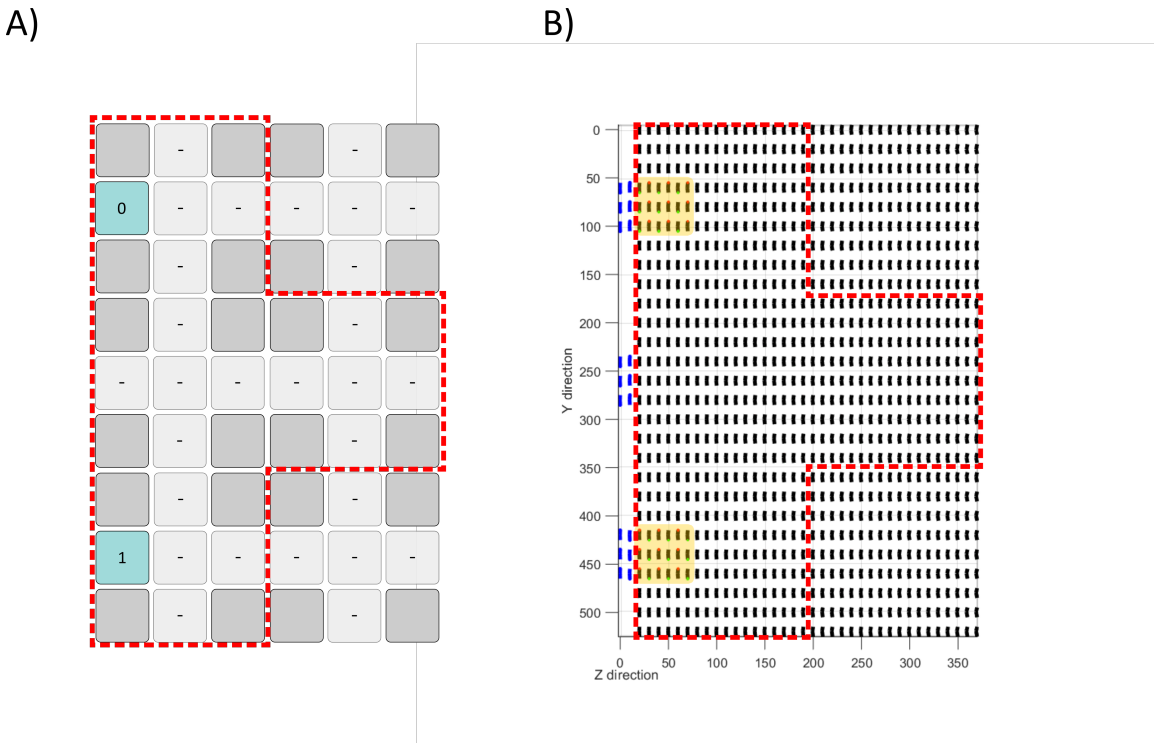


Figure 7.18: Simulation progression - Step 1 (Majority voter)

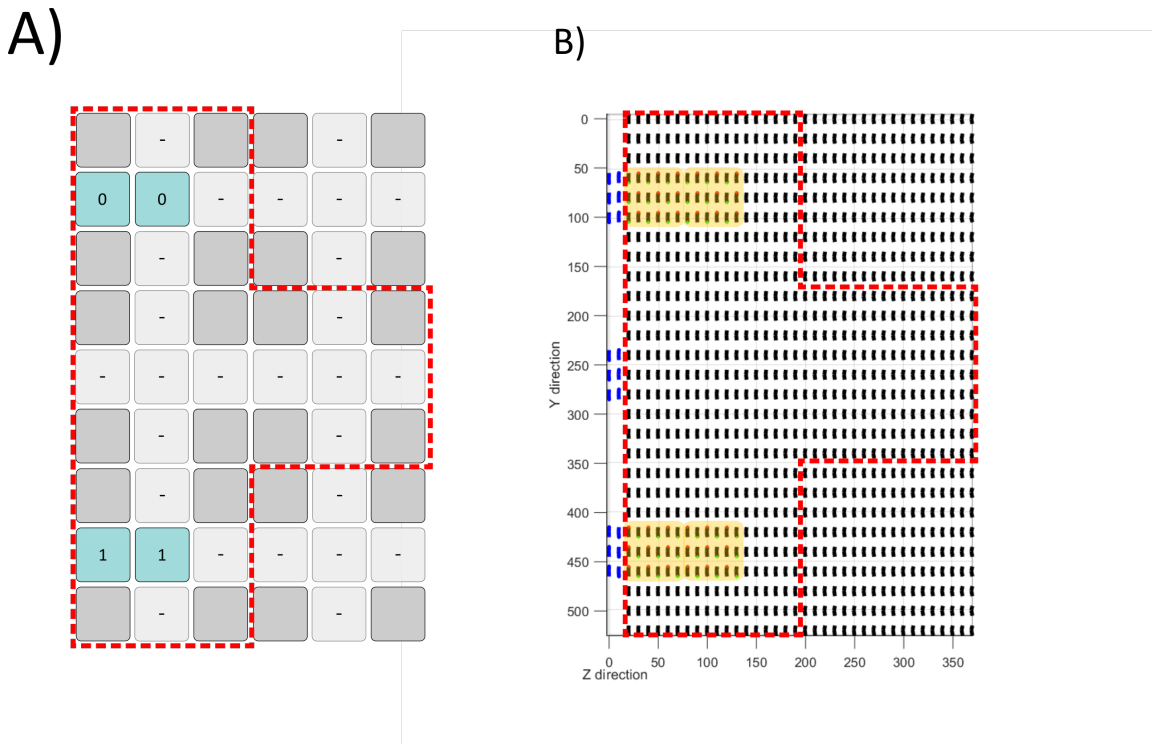


Figure 7.19: Simulation progression - Step 1 (NAND)

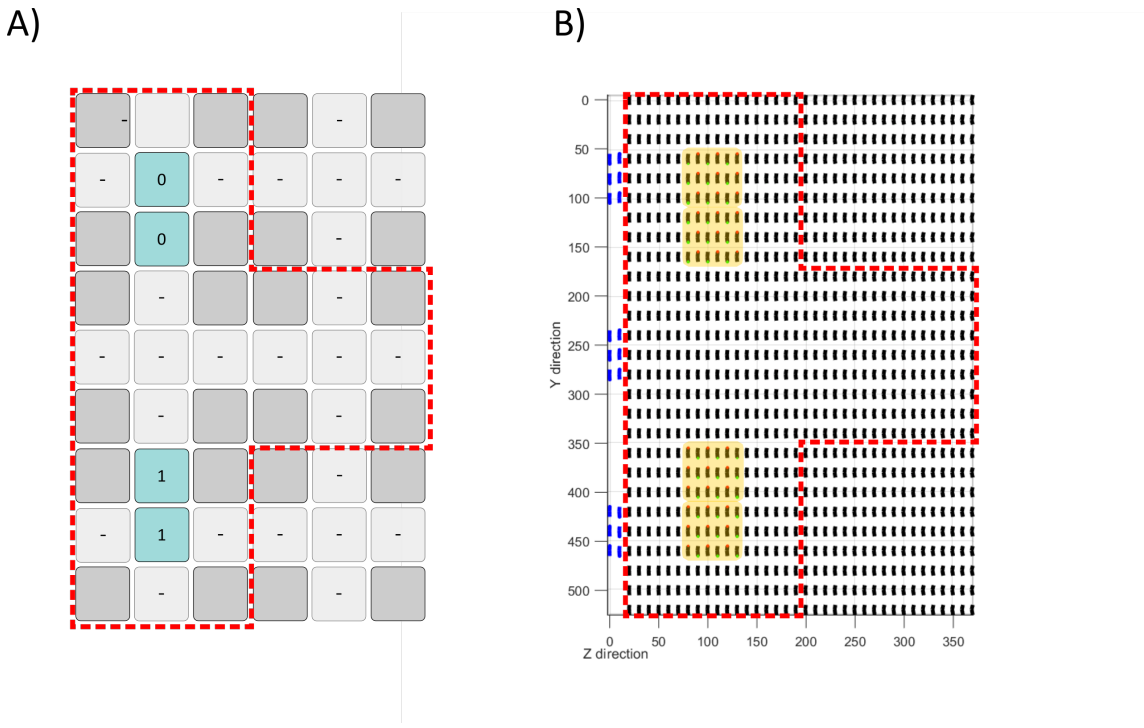


Figure 7.20: Simulation progression - Step 2 (NAND)

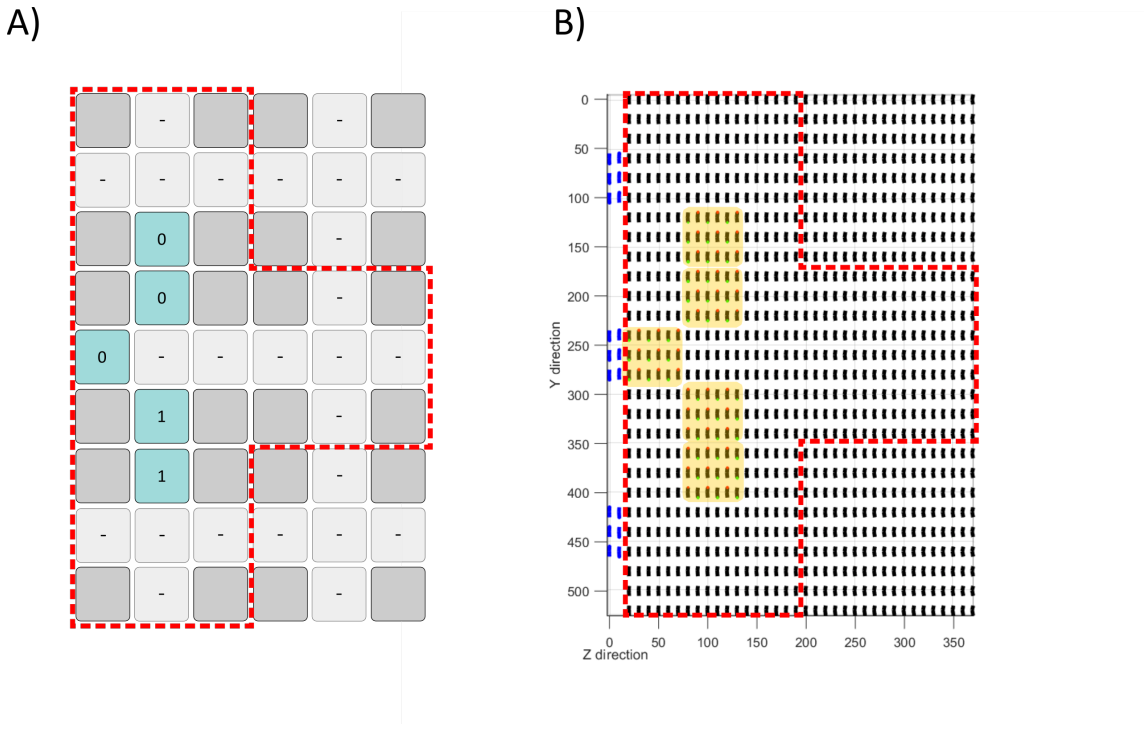
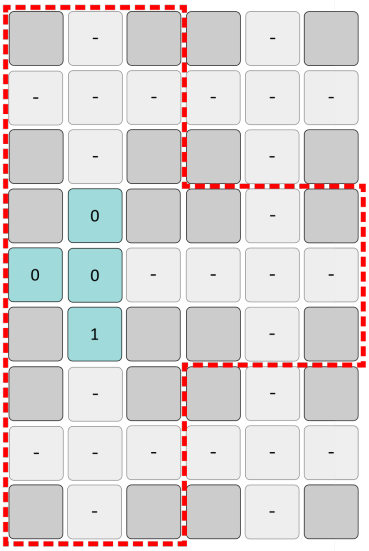


Figure 7.21: Simulation progression - Step 3 (NAND)

A)



B)

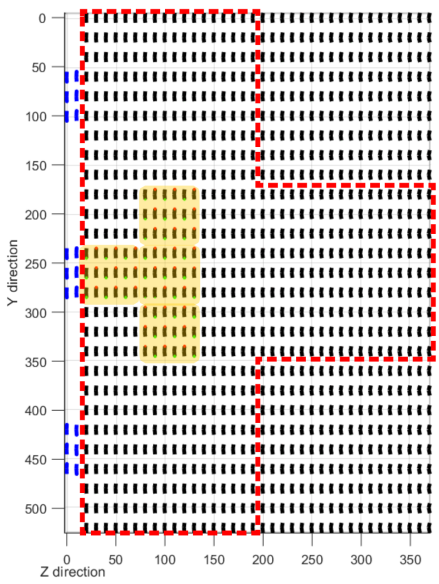
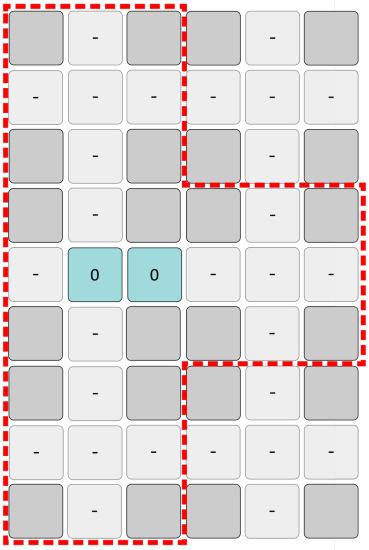


Figure 7.22: Simulation progression - Step 4 (NAND)

A)



B)

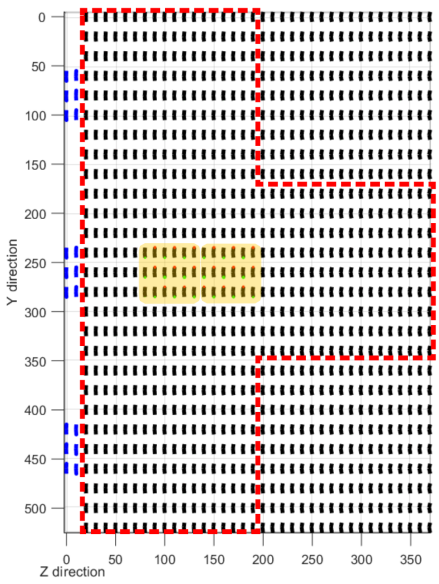


Figure 7.23: Simulation progression - Step 5 (NAND)

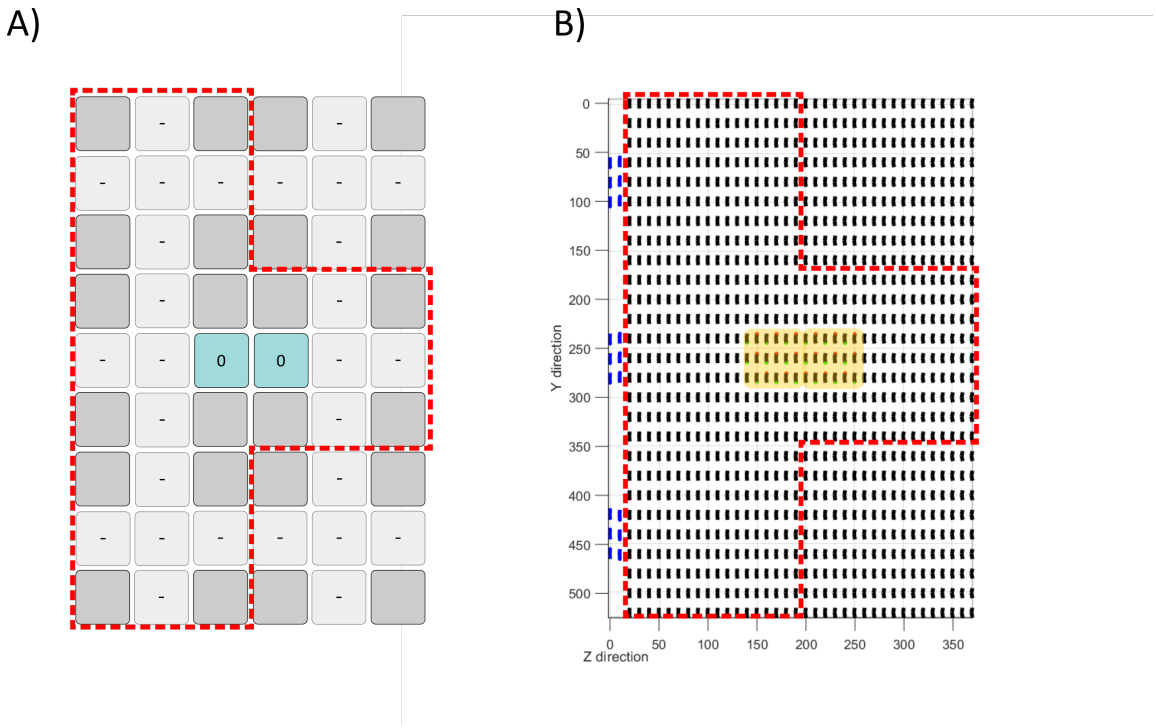


Figure 7.24: Simulation progression - Step 6 (NAND)

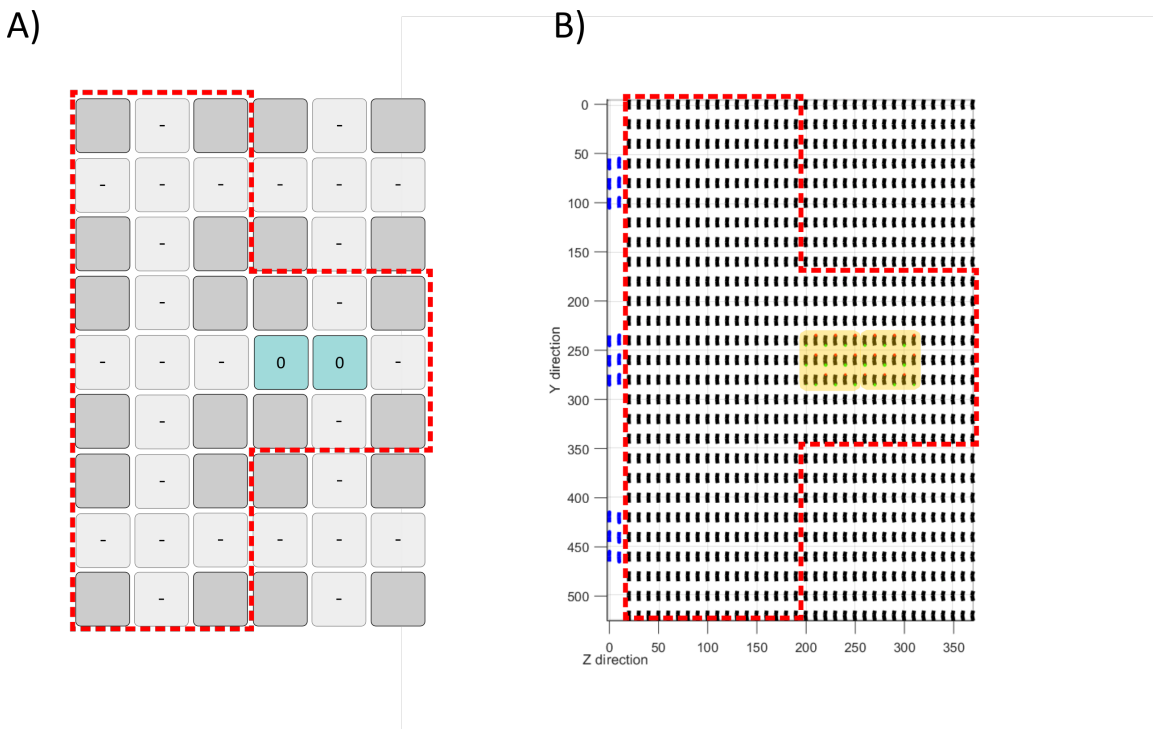


Figure 7.25: Simulation progression - Step 7 (NAND)

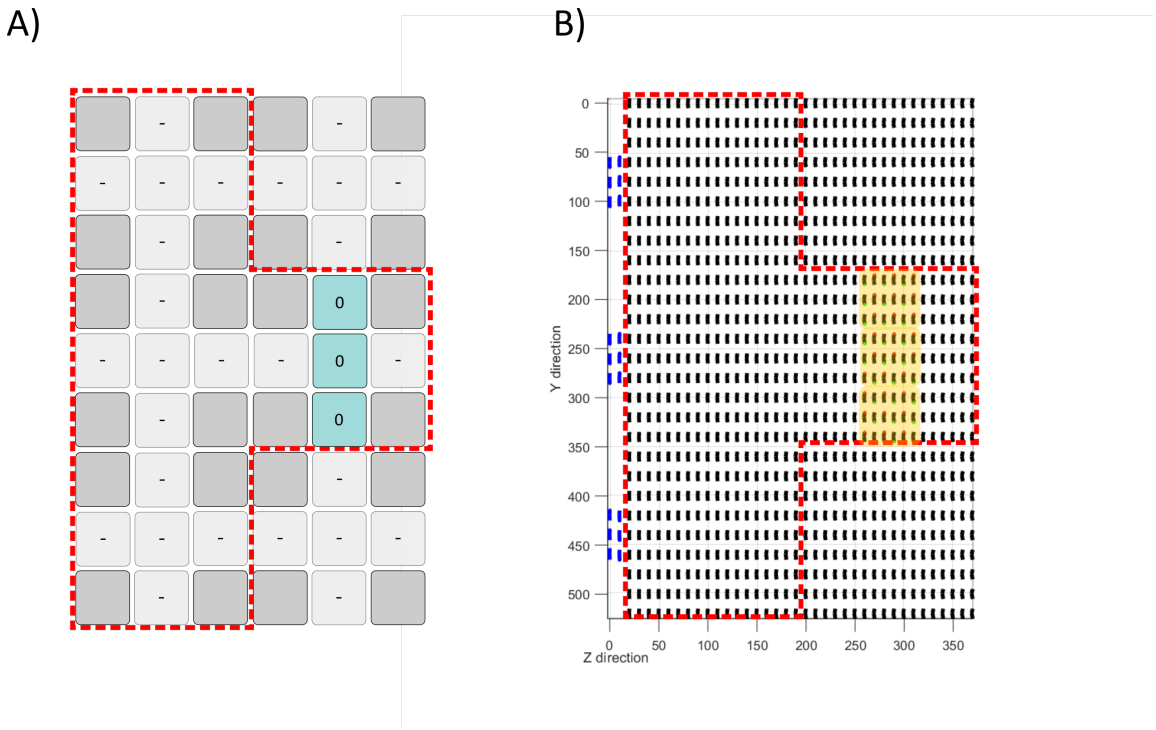


Figure 7.26: Simulation progression - Step 8 (NAND)

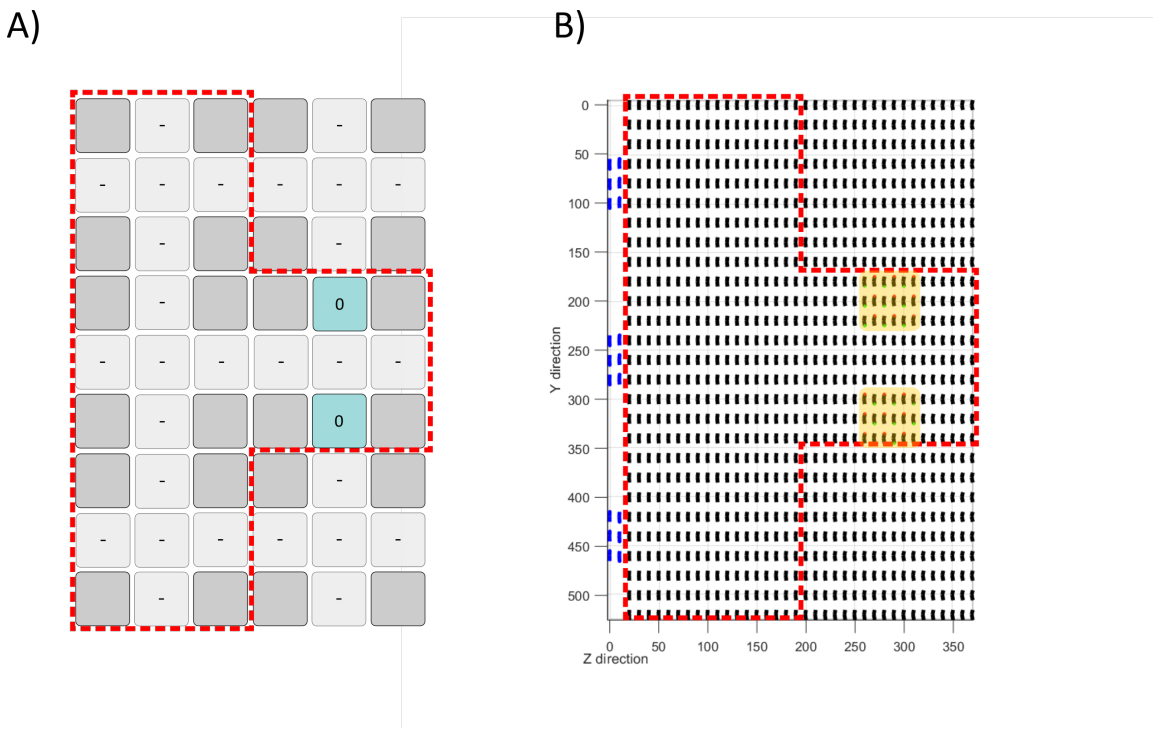


Figure 7.27: Simulation progression - Step 9 (NAND)

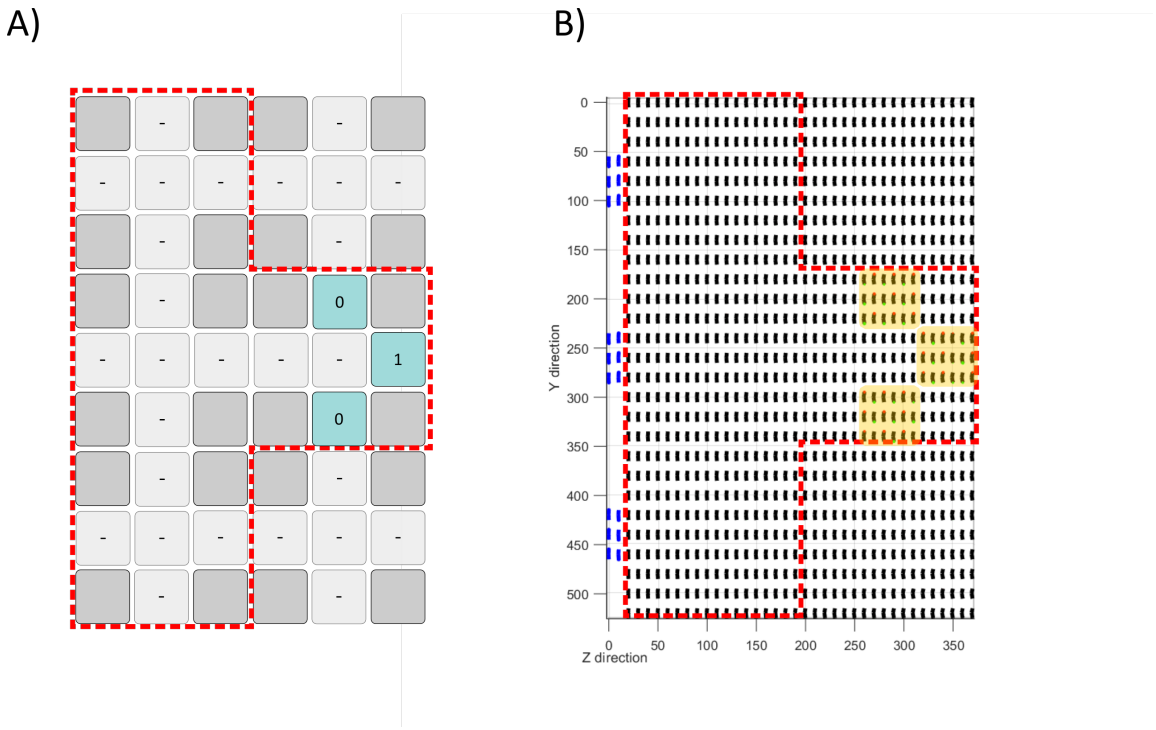


Figure 7.28: Simulation progression - Step 11 (NAND)

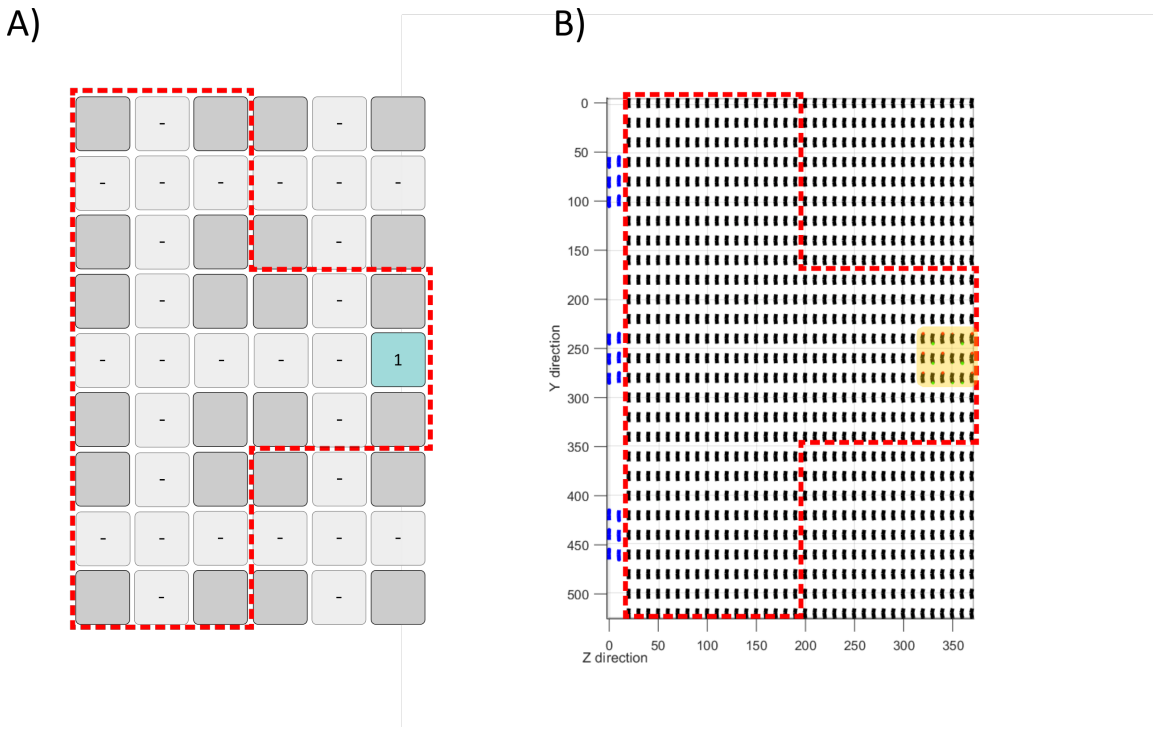


Figure 7.29: Simulation progression - Step 12 (NAND)

7.4 Simulation parameters

When comes to establish the correctness of molecular operations, there are few key parameters that can dramatically modify the outcome:

- Molecular distance
- Block size
- Clock field strength

For both the parameters, the chosen scenario involves the last step of the inversion operation, since it is the most critical. In fact, it is the only configuration in which there is a diagonal interaction between molecular blocks: only few molecules are exposed to each other, making the interaction weaker than the parallel block-to-block interaction. Hence, the sensibility to distance and clock variations is maximum.

7.4.1 Molecular distance

The basic FCN principle takes advantage of charge repulsion. The force exercised by a point charge on the nearby chargers depends strongly, according to Coulomb law, by the distance. When the charge is stored into a molecule, this dependency is transferred to distance that occurs between molecules. What determines the distance between molecules is the *hindrance* of thiol groups [13], i.e. the space around the thiol group that has to be kept free of other molecules. The difference between Molecular FCN and Molecular RFCN is that, in the former, the sensibility of operations depends from the distance in two directions instead of one. The distance used as default in the previous simulations is, according to the expected distance between quantum dots [15]:

$$d_x = 1\,\mathrm{nm}$$
$$d_y = 2\,\mathrm{nm}$$

The following simulations show an attempt to establish what is the maximum perturbation on the distance that keeps allowing the computation free of logic errors. According to the simulations, the minimum and maximum distance for which the interaction is correct is

$$d_{xmin} = 0.8\,\mathrm{nm} \qquad d_{xmax} = 1.1\,\mathrm{nm}$$
$$d_{ymin} = 1.8\,\mathrm{nm} \qquad d_{ymax} = 2.1\,\mathrm{nm}$$

Figure 7.30 shows the boundary values for which it the interaction occurs correctly, while fig. 7.31 displays two distance values for which there output of majority voter is opposed to what expected. It can be noticed how the shortest distance enlarges the molecular charge dots, enclosing a bigger charge insite the redox center.

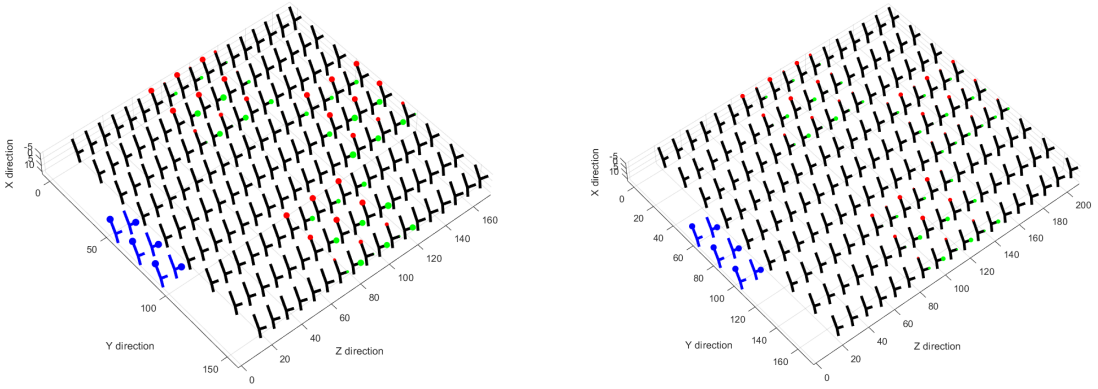


Figure 7.30: Correct operation boundaries. Left simulation: d_{min} = (0.8, 1.8) nanometers. Right simulation: d_{max} = (1.1, 2.1) nanometers.

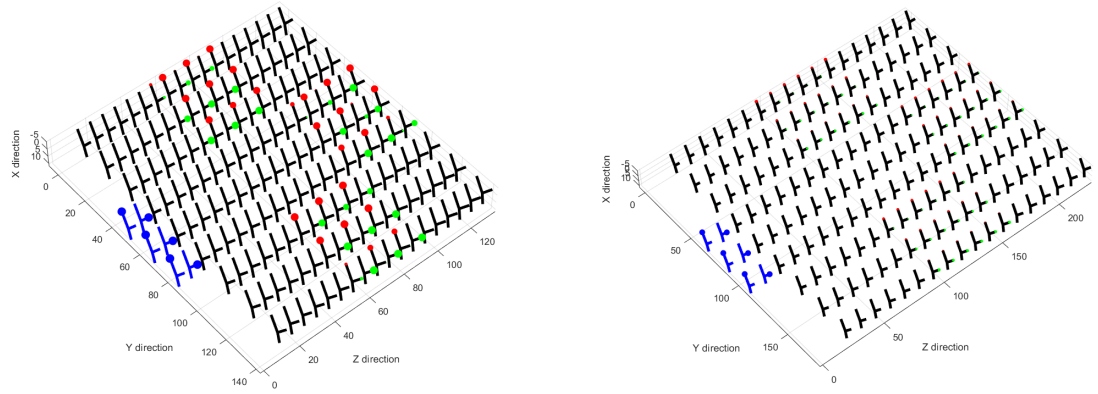


Figure 7.31: Incorrect operation occurred. Left simulation: d = (0.7, 1.7) nanometers. Right simulation: d = (1.2, 2.2) nanometers.

7.4.2 Block size

Each set of molecules subject to the same clock field experiences a feedback effect, which is calculated through the self-consistent algorithms [16]. The block size is then a parameter that can influence the correctness of the logic operation for two reasons. The first scenario is that a small block hasn't enough molecules for the feedback to take place, resulting into the block not being able to hold the logic state, even with a hold field favoring the memorization of the

charge position. In fact, the feedback effect that introduces the *bistability* requires a minimum number of molecules to take place. The second scenario is that a too large block has a feedback effect that is too intense, therefore the nearby block will not be able to interact with it. From this perspective, it can be seen as a *hysteresis* effect. Figures 7.32, 7.33 and 7.34 show how good the hold state can keep the charge stored in the central block. The size that has been simulated is, respectively, $N = 4$, $N = 6$ and $N = 8$. All of these values result into the block having the correct behaviour. Note, however, that the feedback effect is also strongly influenced by the molecular distance, more than by the number of molecules. For the block size simulations, the default distance between molecules is used (dx = 1nm, dy = 2nm).

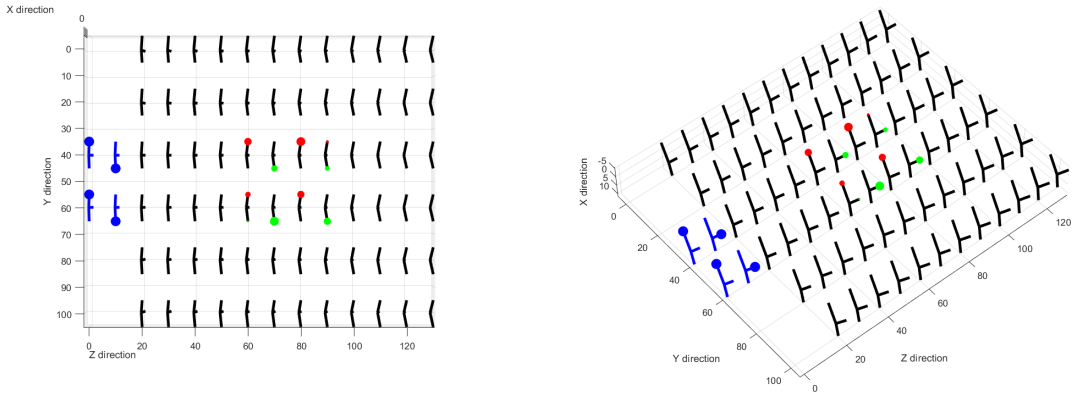


Figure 7.32: Bistability effect - N = 4

7.4.3 Clock field strength

Another factor that influences the ideality of operations is the intensity of the vertical electric field (micro-clock). As seen in section 7.1.2, the trans-characteristic is improved in presence of a positive clock field. The data extrapolated from the atomistic molecular characterization includes the clock field value from a minimum of $-2\,\frac{\mathrm{V}}{\mathrm{nm}}$ (strongest reset state) to a maximum of $2\,\frac{\mathrm{V}}{\mathrm{nm}}$ (strongest hold state). Simulations are performed in order to check the effect of the clock field on the diagonal block-to-block interaction, since it is the worst case scenario. The parametric test is therefore evaluated on the last stage of the inverter, where blocks subject to hold field and to reset field coexist within the standard-cell. The outcome of the tests is shown from fig. 7.36 to 7.38. The figures show the high-level blocks with the corresponding upper electrode voltage, together with a 3D view of the molecular circuit.

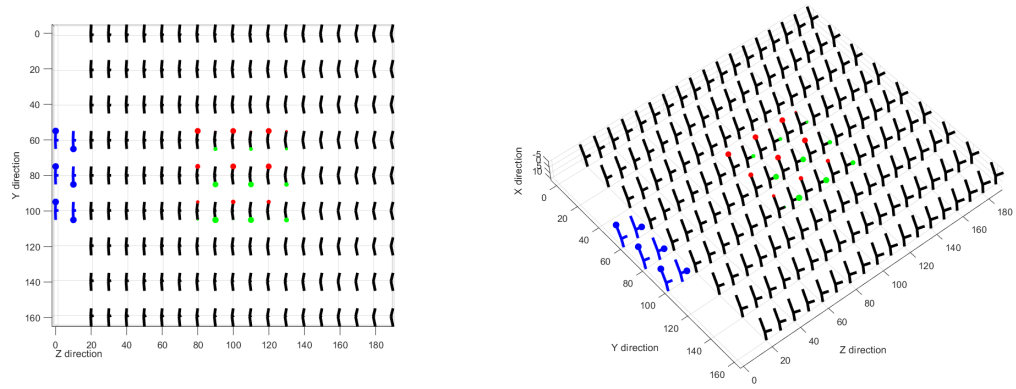


Figure 7.33: Bistability effect - N = 6

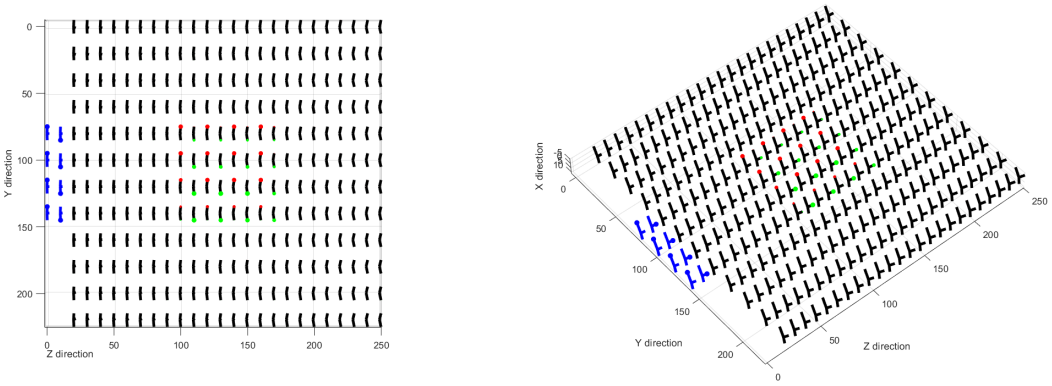


Figure 7.34: Bistability effect - N = 8

Reset field

The parametric test for the reset field is performed while having $2\,\frac{\mathrm{V}}{\mathrm{nm}}$ as default value for the clock field in the hold-state blocks. The first simulation is performed having no reset field at all (0V) in the reset areas. Also the permanent reset on standard-cells corners is kept at 0V. As can be seen from fig. 7.36, there is no distinction between the reset zones and the hold zones, resulting into operation not be able to be performed. Only if the reset field intensity is increased up to $-2\,\frac{\mathrm{V}}{\mathrm{nm}}$, the inverter shows the correct output.

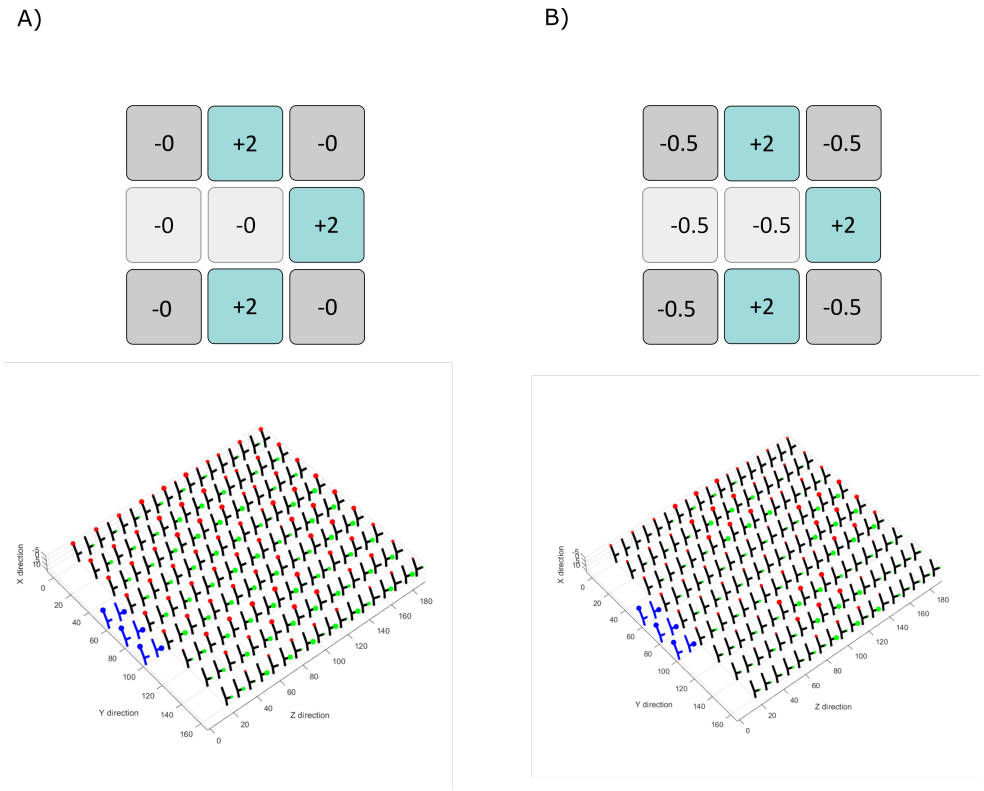


Figure 7.35: Effect of reset field intensity. Value of $0\ \frac{\mathrm{V}}{\mathrm{nm}}$ (left), and $-0.5\ \frac{\mathrm{V}}{\mathrm{nm}}$ (right)

Hold field

The parametric tests for the hold field are performed with a default reset value of $-2\ \frac{\mathrm{V}}{\mathrm{nm}}$. As can be seen from both figures 7.37 and 7.38, the result of the operation is correct even without hold field applied (0V). This is because molecules are only favoured by the positive electric field, but they can perform their task even without. The only constrain is to apply a reset field on all the blocks that aren't supposed to be in hold state. This characteristic opens new possibilities of guiding the operations and the binary propagation, that could then occur by only timing the reset signal instead of both reset and hold. However, since a higher electric hold field increases the amount of charge localized in the dots, it has makes the interaction stronger and more reliable.

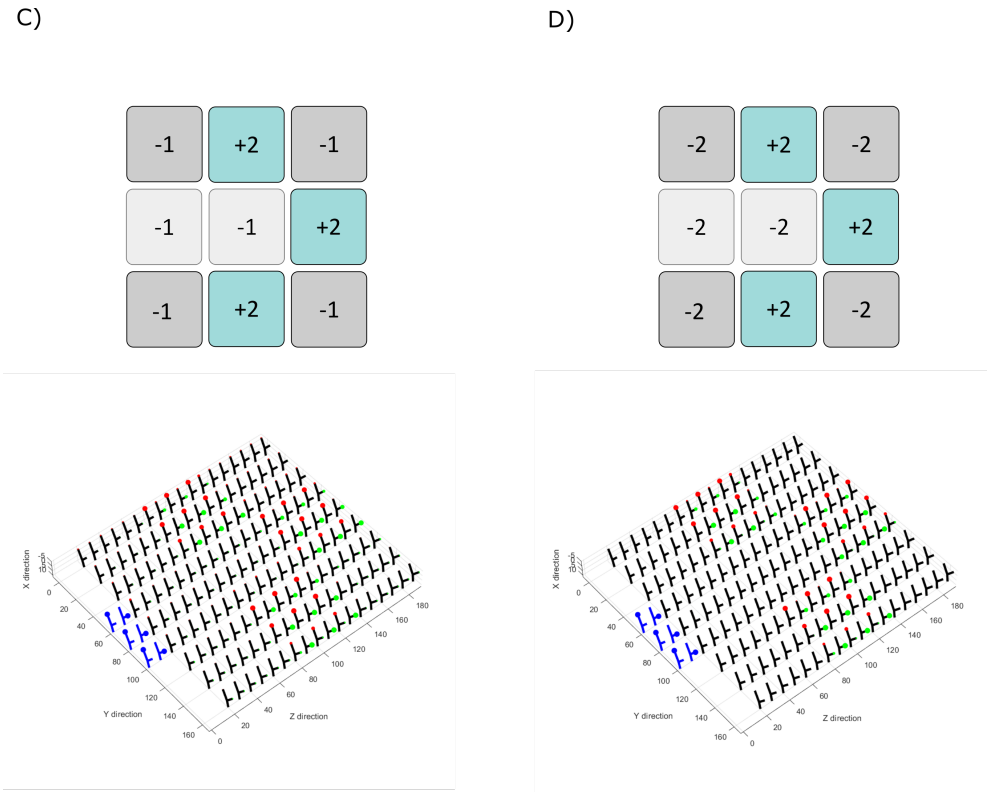


Figure 7.36: Effect of reset field intensity. Value of $-1\ \frac{\mathrm{V}}{\mathrm{nm}}$ (left), and $-2\ \frac{\mathrm{V}}{\mathrm{nm}}$ (right)

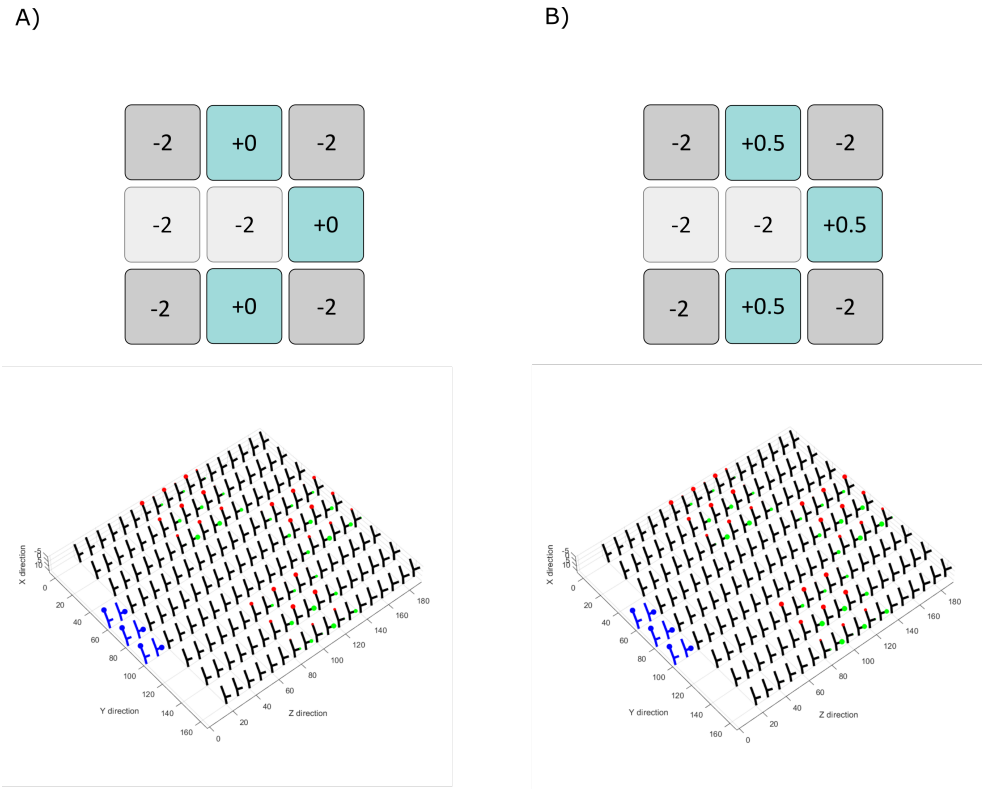


Figure 7.37: Effect of hold field intensity. Value of $0\ \frac{\mathrm{V}}{\mathrm{nm}}$ (left), and $0.5\ \frac{\mathrm{V}}{\mathrm{nm}}$ (right)

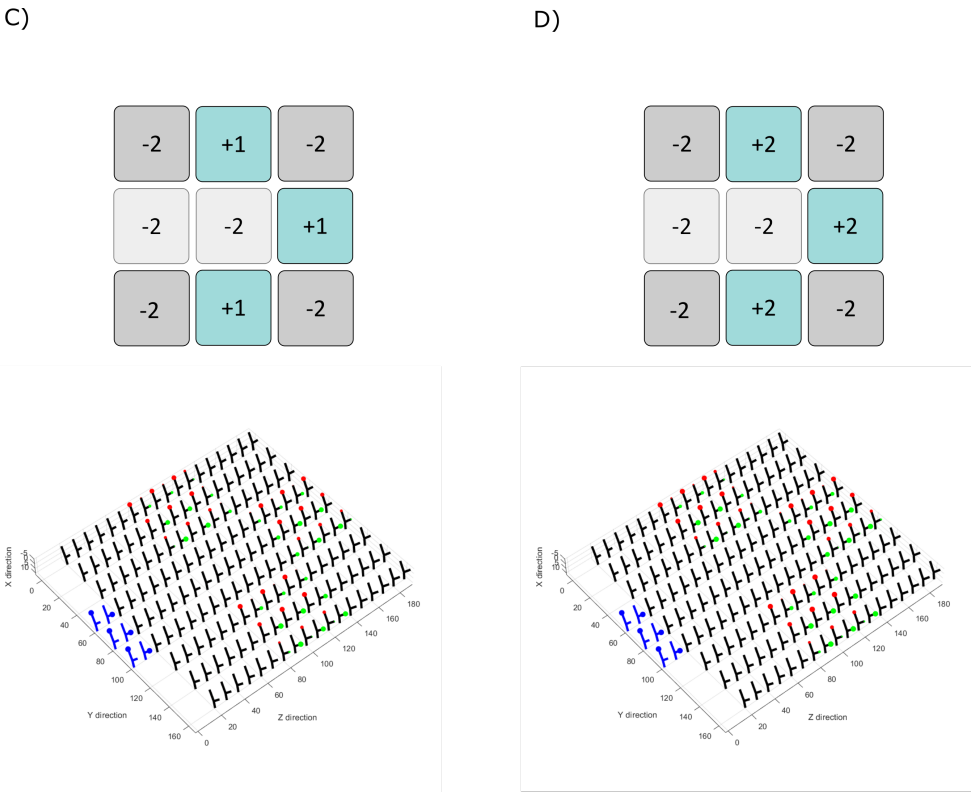


Figure 7.38: Effect of hold field intensity. Value of 1 $\frac{\mathrm{V}}{\mathrm{nm}}$ (left), and 2 $\frac{\mathrm{V}}{\mathrm{nm}}$ (right)

Chapter 8

Analisys of electrical characteristics

This chapters discusses the basic and yet relevant figures of merit of the new RFCN technology. Even if they are strictly related to the particular choice of molecules for the implementation, these figures of merit can be seen as a starting point to frame the new paradigm with respect to CMOS. The examined figures of merit are:

- Area
- Interconnection density
- Capacitance
- Power dissipation

The previous points are modelled, calculated, and compared to the analogue CMOS values that can be found in the latest technology. This comparison gives a first idea of what are the possible advantages and disvantages of Molecular RFCN over CMOS.

8.1 Area

Even if the area isn't the main concern of today's design, it is still a relevant parameter. A new technology should possibly not sacrifice a huge amount of area for the sake of power dissipation. In Molecular RFCN there different areas can be defined. The quantity can be referred to:

- Area of the block and electrode (A_{block})
- Area of the standard-cell (A_{cell})
- Area needed for a logic function (A_{gate})

Each area definition is analyzed in the next sections.

8.1.1 Molecular distance

The smallest processing element that is found in Molecular RFCN is the bisferrocene molecule, which makes the molecular distance the first factor to influence the area. Defining with d_x the distance between two molecules in the longitudinal direction (x) and with d_y the distance of molecules in the lateral direction (y), the following values apply:

$$d_x = 1\,\mathrm{nm}$$
$$d_y = 2\,\mathrm{nm}$$

The distance is set to the the default used for simulations, according to 7.4.1.

8.1.2 Block area

Once the distance is known, the overall area of a block of molecules can be calculated. It is defined as L_x the longitudinal size of the block (x), and as L_y the lateral size of the block (y). Assuming that the block is a square, N molecules will lie on the X axe and N/2 molecules on the Y axe. If the spare distance of the molecules situated at block boundaries is neglected, the following equation applies:

$$L_x \cong N \cdot d_x$$
$$L_y \cong \frac{N}{2} \cdot d_y$$

The area can then be retrieved as

$$A_{block} = L_x \cdot L_y = \frac{N^2}{2} \cdot d_x \cdot d_y$$

Assuming N = 6, the size of the block becomes

$$L_x = 6\,\mathrm{nm}$$
$$L_y = 6\,\mathrm{nm}$$
$$A_{block} = 36\,\mathrm{nm}^2$$

8.1.3 Standard-cell area

Each standard cell is, by construction, made by 9 containers. In order to obtain the area occupied by a cell, the block area has to be multiplied by nine:

$$A_{cell} = 9 \cdot A_{block} = \frac{9}{2} N^2 \cdot A_{base}$$

Substituting the value of the block size, the area of the cell becomes

$$A_{cell} = 324\,\mathrm{nm}^2$$

8.1.4 NAND area

The operator chosen for the area comparison is the NAND gate. Its area can be determined multiplying the standard cell area times *M*, the number of cells required by the operator:

$$A_{NAND} = M \cdot A_{cell} = \frac{9}{2} M N^2 \cdot A_{mol}$$

In order for the NAND gate to properly work, it needs 4 standard cells (of which 3 are required for the AND, and 1 for the inverter). Taking into account also the lateral cells with respect to the inverter, the number of cells required becomes 6, which gives raise to a NAND area of

$$A_{NAND} \cong 2000\,\mathrm{nm}^2$$

The obtained value will be used for the comparison with a FinFET NAND cell.

8.2 Interconnections density

An important aspect to take into account during manufacturing, and that can have big consequences on the electrical characteristics, is the routing of the interconnections: it could negatively affect the overall power balance. In this work a first estimation of interconnections density is made. Currently, in FinFET technology, tools that perform *Place&Route* start from synthesized netlists to perform, thought different algorithms, the design of the interconnections. The routing *congestion*, i.e. the overcrowding of metal density in a certain area, is a direct consequence of a too high FinFET standard-cells density inside the same area. This phenomenon suggests a possible preliminary estimation method for the future congestion within this new FCN technology. A FinFET standard-cell has a layout that exposes a certain number of PINs to the outside world. Those pins will be connected to the metal layers. The method that is used for the estimation compares the density (per unit area) of PINs exposed by a FinFET standard-cell, to the one arising from Molecular block's electrodes. Their ratio will give a first indication of the crowding that interconnections might experience when linked to the PINs. Figure 8.1 compares the number of PINs present, per unit area, in a NAND gate implemented in FinFET technology (7nm) and RFCN. The molecular block size used for the drawing is

$$N = 6$$

Which corresponds to a block area and electrode size of

$$A_{block} = 36\,\mathrm{nm}^2$$

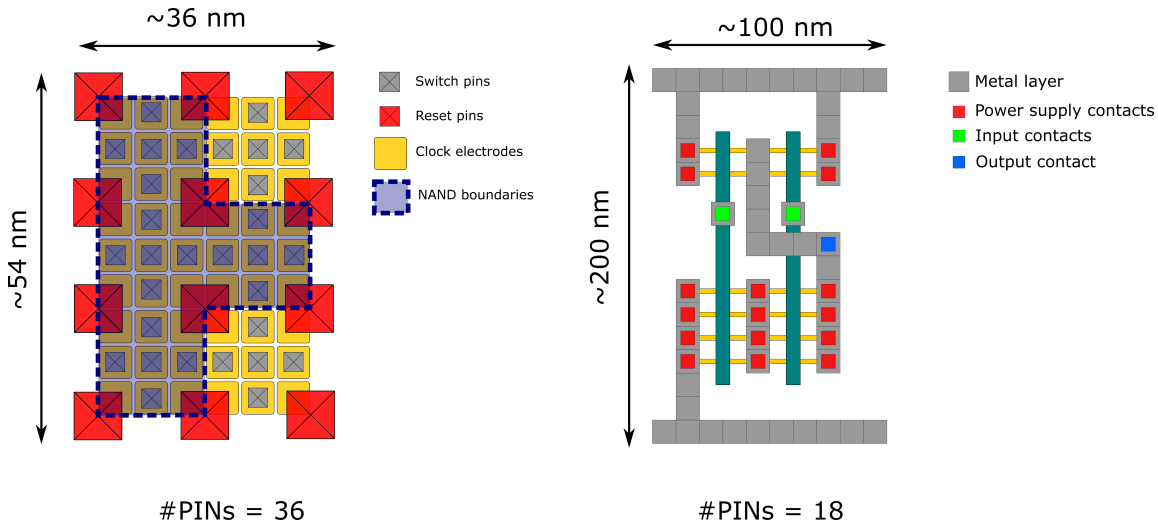


Figure 8.1: Comparison between NAND gate terminals in RFCN layout (left) and FinFET layout (right)

According to the previous figure, the number of PINs per unit area p is:

$$p_{FinFET} = \frac{PIN}{Area} = 900\,\frac{\mathrm{PINs}}{\mu\mathrm{m}^2}$$
$$p_{RFCN} = \frac{PIN}{Area} = 18\,500\,\frac{\mathrm{PINs}}{\mu\mathrm{m}^2}$$

Since a high density of terminals translates into a more challenging metal layers routability, the above numbers suggest that metal wires congestion could be much higher in RFCN, when compared to FinFET.

8.3 Capacitance

The presence of metal electrodes introduces a capacitance between the upper electrodes and the ground plane. Because the electrode defines the block boundaries, the capacitance is linked to

the single block. Hence, the capacitance depends on the area of the block. The capacitance for a parallel plate geometry is given by the following formula:

$$C = \epsilon \cdot \frac{A}{h}$$

Therefore an estimation of the electrodes distance h and the electric permittivity ϵ is required.

8.3.1 Lateral capacitance neglecting

The electrode that lies above molecules faces directly to the ground plane. However, its thickness will add a component of the capacitance towards the nearby electrodes, as shown in 8.2. Nevertheless, for the capacitance estimation, it is neglected, because the planar area is supposed to be much greater than the sum of the lateral sections.

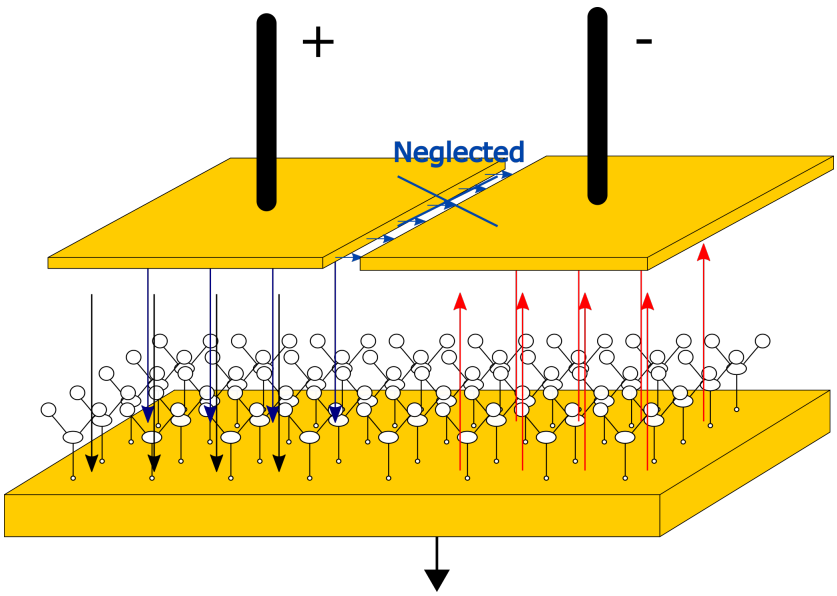


Figure 8.2: Lateral and planar capacitance effects.

As a result, the following simplification applies:

$$C = C_{main} + C_{lateral} \cong C_{main} = C$$

8.3.2 Electrodes distance

In order for the vertical electrical field to act as ideally parallel, the distance between the upper electrode and the ground plane should be such to guarantee no electric field curvature. The height of a bisferrocene molecule is 2 nm [13], therefore a recommended electrodes distance could be of the following size:

$$h = 4\,\mathrm{nm}$$

8.3.3 Electric permittivity

The electric permittivity is given by

$$\epsilon = \epsilon_r \cdot \epsilon_0$$

If the medium between the electrodes is empty, the previous formula becomes

$$\epsilon = \epsilon_{r_{air}} \cdot \epsilon_0$$

However, the presence of molecules disturbs the dielectric response, and introduces a modified dielectric constant:

$$\epsilon = \epsilon_{r_{mol}} \cdot \epsilon_0$$

Nevertheless, for this first approximation analysis the value used is

$$\epsilon_{r_{mol}} \cong \epsilon_{r_{air}}$$

Finally, according to the previously calculated values of area and dielectric constant, the capacitance of the single electrode is

$$C = \epsilon \cdot \frac{A}{h} \cong 0.08\,\mathrm{aF}$$

8.4 Power

One of the reasons that has driven the development of the Field Coupling Nanocomputing and, generally speaking, Quantum-Dot Cellular Automata, has been the theoretical absence of charge flow (current) [2]. The absence of currents could avoid the highest contributions of power dissipation in a chip. In fact, the challenge in today's integrated chips is to obtain the highest *power per performance* ratio, i.e. to minimize the power dissipation. The only hypothetical energy consumption that occurs in FCN and QCA might be related to the displacement of the charge within the container in which the quantum dots belong to. However, the need of a clock network modifies the plot. Since the electrodes experience a capacitive effect, an energy dissipation source might be caused by the signal that load the electrodes. At this point, a scenario is proposed: the electrodes are driven by CMOS logic. In order to have a fast interaction between

the blocks, the electrodes should be loaded and discharged at very high frequencies: a possible solution is then to provide and sink the current through a pullup/pulldown CMOS stage, as shown in fig. 8.3.

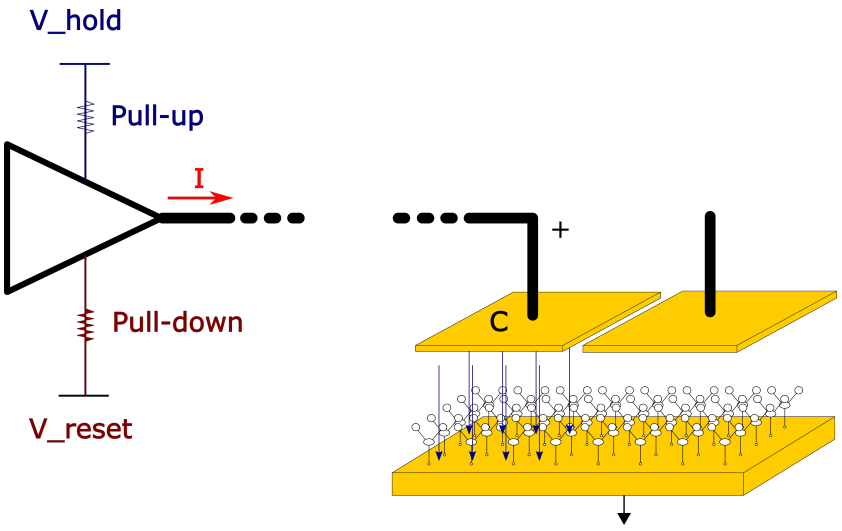


Figure 8.3: Energy is dissipated on pull-up and pull-down resistances of the driving network

The previous hypothesis determines the power dissipation calculations that follows. Note, however, that this power consumption estimation neglects the energy dissipation due to interconnections and drivers static leakage. The energy used to fill the electrodes is for half dissipated on the pull-up resistance, and for another half dissipated on the pull-down resistance. For the estimation, the following hypothesis apply:

a. The power is calculated on a NAND operation

b. Each electrode has a capacitance C

c. Each driver is supplied by a voltage V_{PS}

d. The NAND is executed at frequency f

e. During a NAND, an overall of K electrodes are activated

This means that the dynamic power dissipated expression can partially exploit the CMOS formula

$$P_{NAND} = C \cdot V_{PS}^2 \cdot f \cdot K$$

8.4.1 Power supply voltage

The driver has the task to raise the voltage of the electrode to V_{hold}, in order to bring the molecular block into a hold state. On the other hand, it should load the electrode to $-V_{reset}$ to put the set of molecules in the block into a reset state. This means that the power supply is given by

$$V_{PS} = V_{hold} - V_{reset}$$

The supply voltage applied on the electrode plate generates an electric field. The intensity of the electric field should be high enough displace the charge towards up direction (hold state) and down direction (reset state). As seen in subsection 7.4.3, the value of the electric field in the molecule affects the ideality of the operations. The required minimum electric field to guarantee the correctness of the molecules interaction sets a constrain on the minimum voltage. On the other hand, a too high voltage can affect in a negative manner the power dissipation. The default values for the electric field are

$$E_{hold_{default}} = 2\,\frac{\mathrm{V}}{\mathrm{nm}}$$
$$E_{reset_{default}} = -2\,\frac{\mathrm{V}}{\mathrm{nm}}$$

According to these values, and taking into account the distance between molecules h, the calculated corresponding voltage is

$$V_{hold_{default}} = E_{hold_{default}} \cdot h = 2\,\frac{\mathrm{V}}{\mathrm{nm}} \cdot 4\,\mathrm{nm} = 8\,\mathrm{V}$$

$$V_{reset_{default}} = E_{reset_{default}} \cdot h = -2\,\frac{\mathrm{V}}{\mathrm{nm}} \cdot 4\,\mathrm{nm} = -8\,\mathrm{V}$$

These values lead to a default power supply voltage of

$$V_{PS_{default}} = 16\,\mathrm{V}$$

The calculated power supply voltage could, however, be hardly handled in a chip. Not only it would increase exponentially the dynamic power requested by the drivers, but could generate tremendous leakages (which are not taken into account by this analisys).
However, simulations in 7.4.3 show that is still possible to obtain a correct overall behaviour even without a hold electric field, and using the following reset field value:

$$E_{hold} = 0\,\frac{\mathrm{V}}{\mathrm{nm}}$$
$$E_{reset} = -1.5\,\frac{\mathrm{V}}{\mathrm{nm}}$$

Repeating the previous calculations, the power supply voltage needed for this field configuration is

$$V_{PS} = 6\,\mathrm{V}$$

8.4.2 Frequency of operation

The system clock (sCK) takes care of synchronizing the logic operations, and sets frequency with which each is performed. Having no constrains for the frequency, this analisys sets the frequency value close to a common clock frequency found in today's technology. This hypothesis allows for an easier comparison between the power dissipated by a FinFET NAND gate and the one dissipated by an RFCN NAND gate.

$$f = 2.9\,\mathrm{GHz}$$

The maximum frequency of operation isn't subject to this analysis. Having the electrodes a very small capacitance value, a higher frequency could possibly be achieved.

8.4.3 Number of electrodes

A NAND gate in RFCN is made by 4 standard cells, and each standard cell has to activate 5 electrodes during the overall micro-operations, as shown in fig. 7.5. This means that, for a NAND operation, 20 electrodes are totally activated. The number of electrodes *K* is therefore

$$K_{NAND} = 20$$

Using the previous values, the dynamic power calculated for a NAND operating at $2.9\,\mathrm{GHz}$ and with a block size of $N = 6$ results being

$$P_{NAND} = C \cdot V_{PS}^2 \cdot f \cdot K = 166\,\mathrm{nW}$$

8.5 CMOS comparison

The estimation of the three figures of merit is useful to have a first benchmark of the RFCN technology. The comparison is made on a NAND operation, since it is widely used and provides logic completeness [12]. The chosen NAND gate for the FinFET side belongs to a possible implementation in $7\,\mathrm{nm}$ technology node [18].

8.5.1 Area

The first comparison is made between the area of a NAND RFCN operator and the area of a FinFET NAND standard-cell. Figure 8.4 displays the variation of RFCN NAND area when increasing the block size, i.e. the number of molecules that lie in the longitudinal direction of the block (*N*). The trend is compared with a FinFET NAND standard-cell, that has a value of

$$A_{FinFET} = 20\,000\,\mathrm{nm}^2$$

The default value highlighted in the graph indicates the area of RFCN NAND using $N = 6$, corresponding to

$$A_{RFCN} = 2000\,\mathrm{nm}^2$$

By looking at this numbers, Molecular RFCN technology brings a reduction of the area occupated by the logic gate of 90%, when using the default block size. The graph shows that, even if increasing the block size to $N = 14$ (14x7 molecules) the area occupation with respect to CMOS is still the half.

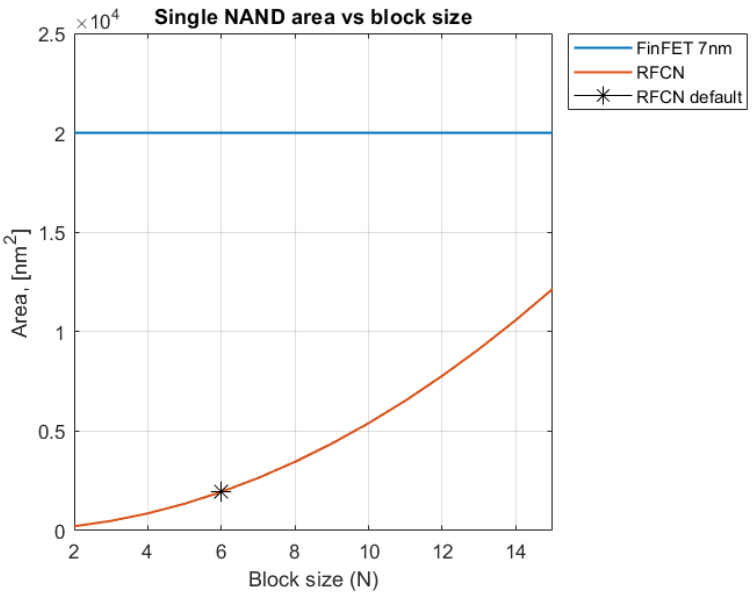


Figure 8.4: NAND area occupation: RFCN vs FinFE

8.5.2 Dynamic power

The second comparison is made between the dynamic power dissipated by a NAND RFCN operator and a FinFET NAND standard-cell operating at $2.9\,\mathrm{GHz}$. Figure 8.5 displays the variation of NAND dynamic power when increasing the molecular block size.

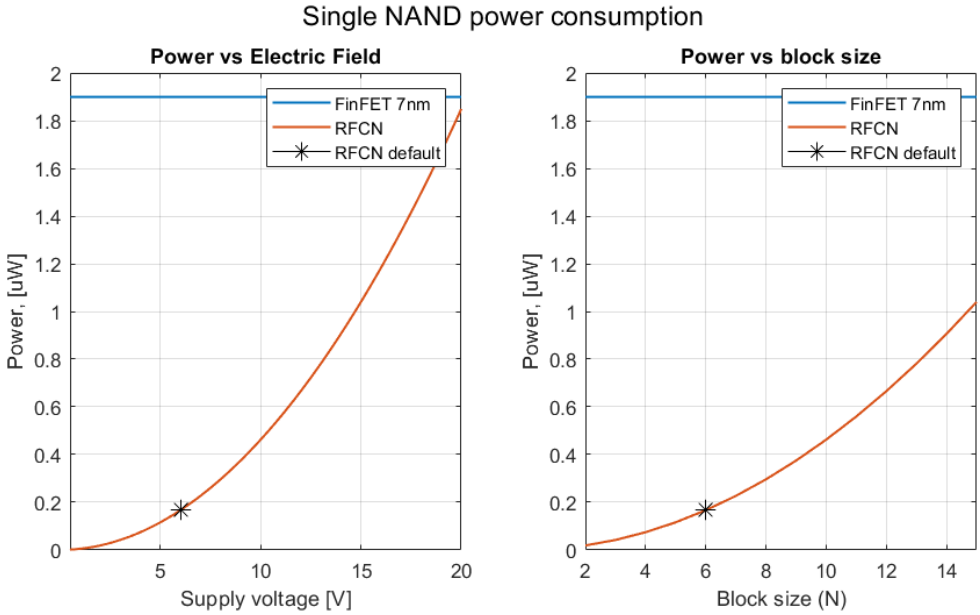


Figure 8.5: NAND dynamic power dissipation @ $2.9\,\mathrm{GHz}$: RFCN vs FinFET

The trend is compared with a FinFET NAND gate. The FinFET power dissipation is calculated on a frequency of $2.9\,\mathrm{GHz}$, considering an activity factor of 100%. The value is

$$P_{FinFET} = 1.9\,\mu\mathrm{W}$$

The default value highlighted in the graph indicates the power dissipation of RFCN NAND calculated in the previous sections:

$$P_{RFCN} = 166\,\mathrm{nW}$$

It can be observed how RFCN technology brings also reduction of the dynamic power dissipation by the logic gate of 90% on the NAND cell. The graph in fig. 8.5 displays how the power increases quadratically when increasing the supply voltage, due to the higher dissipation on the pull-up/pull-down resistances of the clock network. Also the block size plays a role into the increase of the power dissipation: when enlarging the electrodes, a higher capacitance is to be loaded by the drivers.

8.5.3 Interconnections

The third comparison tries to predict how the interconnections' congestion behaves when modifying the molecular block size. Also a comparison with the PIN density of the NAND FinFET cell is made. Figure 8.6 displays the variation of RFCN NAND terminals density when varying the block size.

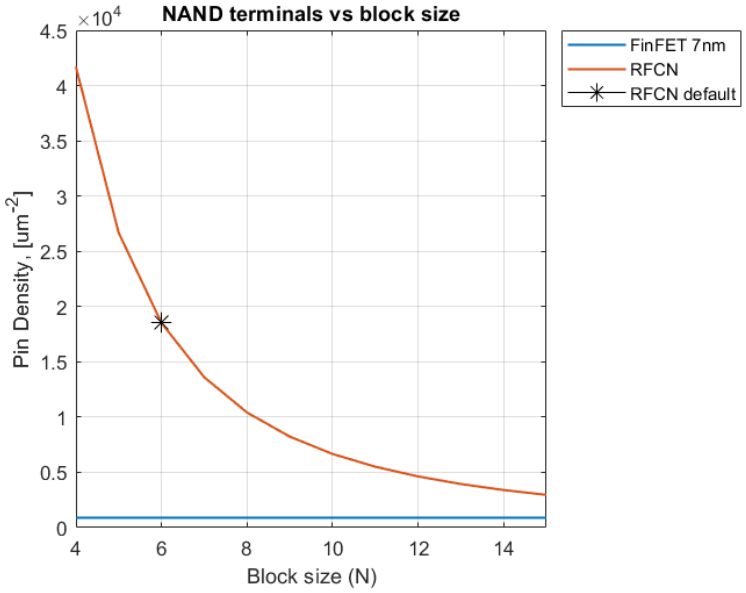


Figure 8.6: NAND pin density per unit area: RFCN vs FinFET

According to the estimation in section 8.2, the PIN density of a RFCN NAND gate with $N = 6$ corresponds to

$$PIN_{RFCN} = \frac{PIN}{Area} = 18\,500\,\frac{\text{PINs}}{\mu\text{m}^2}$$

While the calculated PIN density in CMOS technology for a NAND gate is

$$PIN_{FinFET} = \frac{PIN}{Area} = 900\,\frac{\text{PINs}}{\mu\text{m}^2}$$

From fig. 8.6 it can be observed how the congestion value of RFCN gets closer to CMOS only when the block size has $N > 14$. However, the estimation doesn't take various factors into account that could favor RFCN over FinFET. FinFETs are devices that exploit electrical currents in order to perform operations, therefore the interconnections should be robust enough to supply the high current requested. Therefore, they are very large (to minimise non-ideal

effects such as *IR-Drop*). RFCN, instead, does not need power supply: its only connections are towards the clocking network. Since the current flowing into the clock electrodes has the task to supply very low current to very low capacitances (as shown in section 8.3), the size of the wires can be much smaller. This suggests that RFCN can afford a higher density of metal layers.

8.6 Summary

Exception made for the metal wires estimation, the area and power dissipated by RFCN is strongly favored over CMOS. Nevertheless, since the terminals density of the cell is much higher, the price to pay is a possible crowding of the interconnections. The metal layers congestion can be reduced by increasing the block size, which reflects a higher power dissipation and area occupation, as shown in fig. 8.7: a trade-off must be evaluated according to the desired application.

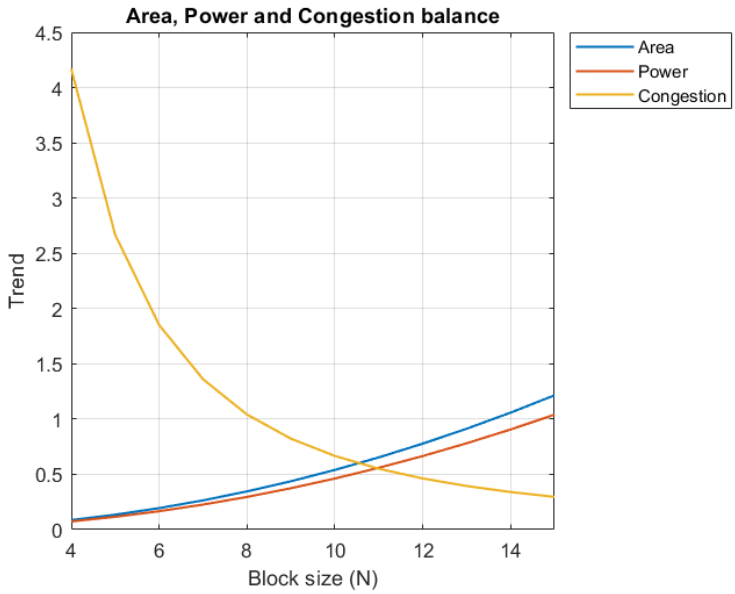


Figure 8.7: Trend: increase of the block size favours the congestion at the expense of power dissipation and area occupation

Chapter 9

Application fields and future development

The new method of computation introduced with Reconfigurable Field-Coupled Nanocomputing proposes a different procedure for data manipulation and circuit design. The choice of using CMOS drivers to activate the electrodes, on the other hand, makes this technology still dependent by the current silicon. The introduction of the drivers represents the biggest source of power dissipation. Nevertheless, a new method for powering the electrodes could bring a further reduction of the energy consumption (which is already lower than CMOS). In any case, the electrical characteristics aren't considered the main advantage of this technology: the improvements lies on another abstraction level: it is represented by the ability to reconfigure the logic and the data pathway using electric signals. Since this technology belongs to "Beyond CMOS" group, it is unlikely that RFCN could replace CMOS in a conventional computer architecture: the new approach could be exploited for new architectures, which are maybe precluded to CMOS logic. Among the application fields, it is believed that the following paradigms could take advantage of the RFCN method:

- Logic-in-memory
- Neural networks
- FPGA

A future development of RFCN could investigate deeper modeling of the system-level clock structure: among the open issues include, in fact, is included the managing of the sequencers and their physical localization. Figure 9.1 displays a possible scheme for abstraction levels in RFCN, in which the *architectural* box is left open for future developments.

Future development could include a design style that takes advantages of the memory islands. However, a more complex logic function than the one reported in this work (XOR) could be performed in many different ways, based on the base building blocks provided: the goal of the research was to not constrain, bind and limit in any manner the potential design styles that RFCN unlocks.

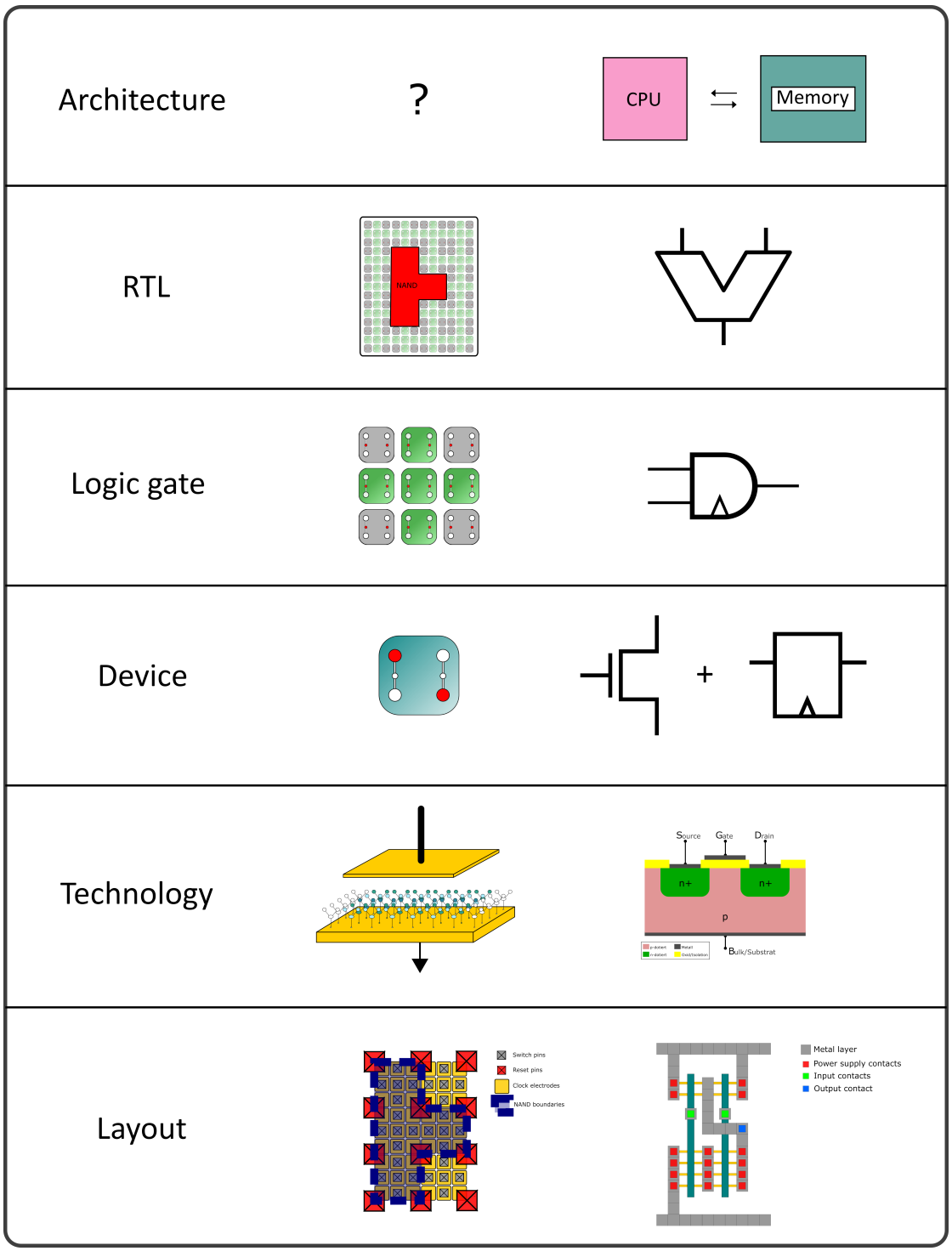


Figure 9.1: RFCN full hierarchical view

Acronyms

ALD Atomic Layer Deposition.

CCL Containers Common Layer.

CMOS Complementary Metal-Oxide-Semiconductor.

DIBL Drain Induced Barrier Lowering.

FCN Field Coupled Nanocomputing.

FPGA Field Programmable Gate Array.

MoSQuiTo Molecular Simulator Quantum-Dot Cellular Automata Torino.

MUT Molecule Under Test.

QCA Quantum-Dot Cellular Automata.

RFCN Reconfigurable Field Coupled Nanocomputing.

SAM Self-Assembled Monolayer.

SCE Short Channel Effects.

SCERPA Self-Consistent Electrostatic Potential Algorithm.

VLSI Very Large Scale Integration.

Bibliography

[1] R. Courtland, "Transistors could stop shrinking in 2021," *IEEE Spectrum*, vol. 53, no. 9, pp. 9–11, 2016.

[2] C. S. Lent, P. D. Tougaw, and W. Porod, "Quantum cellular automata: the physics of computing with arrays of quantum dot molecules," in *Proceedings Workshop on Physics and Computation. PhysComp '94*, pp. 5–13, 1994.

[3] A. Pulimeno, M. Graziano, A. Antidormi, R. Wang, A. Zahir, and G. Piccinini, *Understanding a Bisferrocene Molecular QCA Wire*, pp. 307–338. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[4] G. Sanclemente, "Analysis and simulation of the sinusoidal clocking field in molecular fcn." April 2019.

[5] R. W. Johnson, A. Hultqvist, and S. F. Bent, "A brief review of atomic layer deposition: from fundamentals to applications," *Materials Today*, vol. 17, no. 5, pp. 236 – 246, 2014.

[6] J. K. Bhattarai, D. Neupane, V. Mikhaylov, A. V. Demchenko, and K. J. Stine, "Self-assembled monolayers of carbohydrate derivatives on gold surfaces," in *Carbohydrate* (M. Caliskan, I. H. Kavakli, and G. C. Oz, eds.), ch. 4, Rijeka: IntechOpen, 2017.

[7] V. Sharma, M. Pattanaik, and B. Raj, "Onofic approach: low power high speed nanoscale vlsi circuits design," *International Journal of Electronics*, vol. 101, no. 1, pp. 61–73, 2014.

[8] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernstein, "Quantum cellular automata," *Nanotechnology*, vol. 4, pp. 49–57, jan 1993.

[9] C. S. Lent, B. Isaksen, and M. Lieberman, "Molecular quantum-dot cellular automata," *Journal of the American Chemical Society*, vol. 125, no. 4, pp. 1056–1063, 2003. PMID: 12537505.

[10] Y. Ardesi, L. Gnoli, M. Graziano, and G. Piccinini, "Bistable propagation of monostable molecules in molecular field-coupled nanocomputing," pp. 225–228, 07 2019.

[11] V. Arima, M. Iurlo, L. Zoli, S. Kumar, M. Piacenza, F. Della Sala, F. Matino, G. Maruccio, R. Rinaldi, F. Paolucci, M. Marcaccio, P. G. Cozzi, and A. P. Bramanti, "Toward quantum-

dot cellular automata units: thiolated-carbazole linked bisferrocenes," *Nanoscale*, vol. 4, pp. 813–823, 2012.

[12] H. Enderton, *A Mathematical Introduction to Logic*. Elsevier Science, 2001.

[13] M. Graziano, A. Pulimeno, R. Wang, X. Wei, M. R. Roch, and G. Piccinini, "Process variability and electrostatic analysis of molecular qca," *J. Emerg. Technol. Comput. Syst.*, vol. 12, Sept. 2015.

[14] J. C. Love, L. A. Estroff, J. K. Kriebel, R. G. Nuzzo, and G. M. Whitesides, "Self-assembled monolayers of thiolates on metals as a form of nanotechnology," *Chemical Reviews*, vol. 105, no. 4, pp. 1103–1170, 2005. PMID: 15826011.

[15] M. Graziano, R. Wang, M. Ruo Roch, Y. Ardesi, F. Riente, and G. Piccinini, "Characterisation of a bis-ferrocene molecular qca wire on a non-ideal gold surface," *Micro Nano Letters*, vol. 14, pp. 22–27, 01 2019.

[16] Y. Ardesi, R. Wang, G. Turvani, G. Piccinini, and M. Graziano, "Scerpa: A self-consistent algorithm for the evaluation of the information propagation in molecular field-coupled nanocomputing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2749–2760, 2020.

[17] Y. Ardesi, A. Pulimeno, M. Graziano, F. Riente, and G. Piccinini, "Effectiveness of molecules for quantum cellular automata as computing devices," *Journal of Low Power Electronics and Applications*, vol. 8, p. 24, Jul 2018.

[18] T. Cui, Q. Xie, Y. Wang, S. Nazarian, and M. Pedram, "7nm finfet standard cell layout characterization and power density prediction in near- and super-threshold voltage regimes," in *International Green Computing Conference*, pp. 1–7, 2014.