POLITECNICO DI TORINO

Department of Electronics and Telecommunications
Master Degree in Electronic Engineering

Master Thesis

Design of neural networks based on molecular Field-Coupled Nanocomputing

Supervisors
prof. Mariagrazia Graziano
prof. Gianluca Piccinini
M.Sc. Giuliana Beretta
M.Sc. Yuri Ardesi

Candidate
Federico Ravera

Anno Accademico 2021-2022

"And how many years can some people exist before they're allowed to be free? The answer is blowin' in the wind"

-Bob Dylan

Alla mia nonna, alla quale il tempo non ha permesso di esserci, ma il cui affetto nel vento ha sempre soffiato

Abstract

Research related to Artificial Neural Networks (ANN) enhanced the realization of efficient hardware structures able to perform new and highly complex tasks. The hardware realization of ANN requires massive resources, leading to space requirements, computational time, and power consumption challenges. In parallel to this, in recent years, the scaling process dictated by Moore's law has started to show future limitations related to the same, previously mentioned, technological aspects. Therefore, researchers began to study and develop new ways of realizing digital electronics. Among the range of proposed alternatives, Field-Coupled Nanocomputing (FCN) is one of the most promising in the group of "Beyond CMOS" technologies. The two main characteristics of this paradigm are the possibility of implementing devices in highly dense functional arrays and the reduced power dissipation.

The proposed hardware neural network has its basis in the molecular implementation of the FCN paradigm. In this technology, the information is encoded in the charge distribution within each molecule, whereas electrostatic interactions between neighboring molecules ensure information propagation, thus avoiding current flow. Molecular FCN has shown promising results concerning the realization of standard digital gates.

The work developed in this thesis has its basis on the results already present in the literature, which show the potential advantages of using FCN to implement single neuron models. These results come from the demonstrated linear behavior of the adopted molecules, which are capable of adding up the effects from the surrounding circuit. Starting from these concepts, this thesis expands the analysis towards realizing new hardware architectural solutions for neural network development.

The procedure followed during this work analyzes possible strengths and issues the proposed implementation presents and consists of a two-step analysis of the main structures present in the network. First, the interface molecules have been characterized to encode a certain weight, enlarging their study to cover process variations.

Then, the properties of the output molecules and the information propagation structure have also been described and characterized. Therefore, it was possible to determine the properties the molecules should have to accomplish the task as neurons and derive the layout and timing requirements of the circuit. A Self-Consistent ElectRostatic Potential Algorithm (SCERPA) is adopted to simulate the circuits and solve molecular interactions iteratively.

The outcomes confirm the possibility for the proposed structure to work as a neural network. Indeed, the output of each neuron switches when the sum of the weighted input reaches a defined threshold and the computed pieces of information propagate correctly through the network. Therefore, it was possible to predict the circuit's final output value with remarkable accuracy.

Finally, we demonstrate the functionality of the proposed network to perform pattern recognition tasks and compare the outcomes with those obtained from software-trained feed-forward neural networks. The comparison was successful, and the proposed circuit correctly classified four different 3x3 matrix paths. Moreover, the molecular FCN network uses fewer neurons to perform the task. Furthermore, from a hardware point of view, the structure obtained is compact compared to state-of-the-art silicon solutions, and the involved paradigm avoids current flow. In conclusion, the work developed in this thesis demonstrates the initial expectations: molecules can be used to create artificial neurons, and the proper connection among them can correctly implement working neural networks. This thesis places grounding rules for realizing molecular FCN neural networks; further analysis would be needed to fulfill possible more complex requirements than the ones analyzed so far.

Ringraziamenti

Tengo a ringraziare la professoressa Graziano e il professor Piccinini per avermi dato l'opportunità e la fiducia nello svolgere questo lavoro. Un ringraziamento sentito va anche Giuliana Beretta e Yuri Ardesi, i quali hanno saputo accompagnarmi e consigliarmi sempre puntualmente e con dedizione per tutto lo svolgimento del lavoro di tesi.

Le persone che dovrei ringraziare sono forse troppe per queste poche righe. Parto innanzitutto dal ringraziare profondamente i miei più stretti compagni di studio e di viaggio: Simone, Simone e Miriam. Vi sono grato per la vostra amicizia e supporto in tutti i momenti di questo percorso che abbiamo avuto la fortuna di svolgere, anche se solo parzialmente, insieme. Colgo anche l'occasione di ringraziare Giada per averci sempre sostenuto e tifato per noi. Un sentito ringraziamento va anche a Paolo, grazie per essere sempre stato un fratello per me.

Un grazie anche alle mie sorelle, Erica e Miriam, e ai miei genitori, che sempre silenziosamente mi sono stati vicini e hanno sempre creduto in me.

Infine, il più grande grazie va ad Anna, con la quale tutto ho condiviso e tutto spero di continuare a condividere.

Index

List of figures

List of tables

Chapter 1

Introduction to molecular FCN

1.1 General principles

Information processing followed since the invention of the transistor an unprecedented improvement in the history of technology. Over the past 50 years, the technology evolution in terms of devices miniaturization followed the general trend traced by Moore's law [1]. Moore predicted in 1965 that the number of transistors in an integrated circuit would be doubled every year; actually, in 1975, the prediction was updated to a doubling every two years [2]. The general trend is reported in figure 1.1.

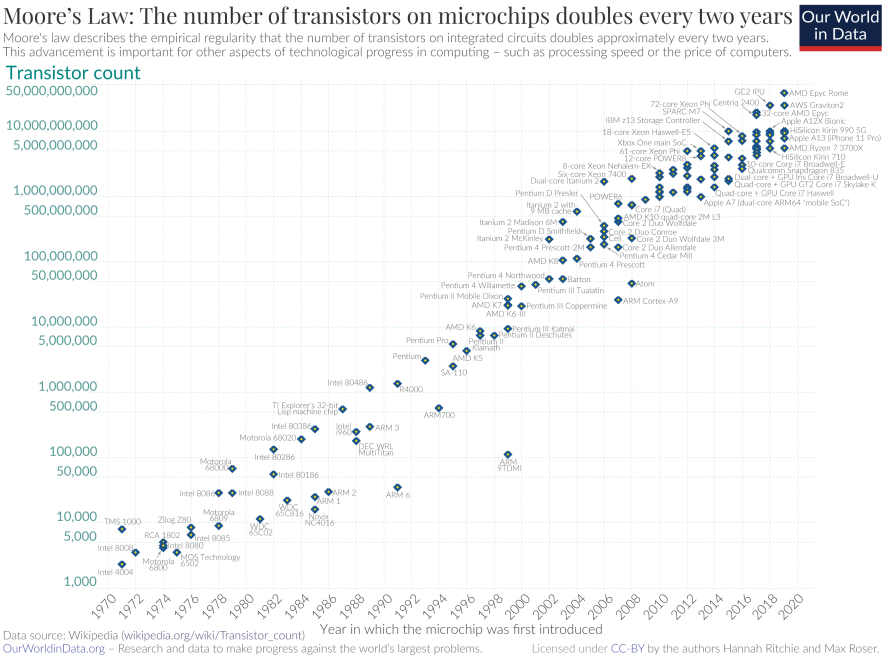
Unfortunately, this proceeding will not always be conceivable. The demand for denser products with reduced power consumption is giving the fabrication companies some problems in realizing such devices. Therefore, silicon technologies seem to show some limitations about those parameters, which are not easy to overcome.

For these reasons, research is moving forward to find solutions to substitute silicon as soon as it becomes unreliable. It is essential to consider that changing a specific technology with a new one is a complex task. Indeed, some primary considerations have to be provided, particularly concerning the standard electronics basic building blocks, as state variables and inputs/output electrical definitions [3] [4].

Among all the characteristics that the candidate technologies must show, power consumption and scalability of devices play a significant role. Moreover, the first one is also related to the speed at which the circuit will work, which increases with short interconnection lines and small devices' dimensions. However, miniaturization involves the presence of more and more elements in a reduced space, meaning that, in some cases, a higher amount of power has to be dissipated. In addition, the increased miniaturization involves possible quantum mechanic effects. These are, for instance, scattering events, tunneling effect, and quantization of energy levels [5].

All these aspects are constantly considered and updated in the ITRS report [6].

Two main paths have been highlighted as possible candidates for the future of integration in electronics [6]:


Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.
This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.
Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

Fig. 1.1: Moore law trend: increase of number of transistors through the years

- More than Moore: this refers to the possibility of producing heterogeneous integrated systems such as System-on-Chip. The idea is to build complex systems integrating all the elements on the same chip.
- Beyond CMOS: in this case, new technologies rely on different paradigms introduced to overcome silicon limitations.

Among all the possible solutions introduced in the ITRS report, the more interesting for this work is the Field Coupling Nanocomputing (FCN) paradigm, implemented in the Quantum Dot Cellular Automata (QCA) technology.

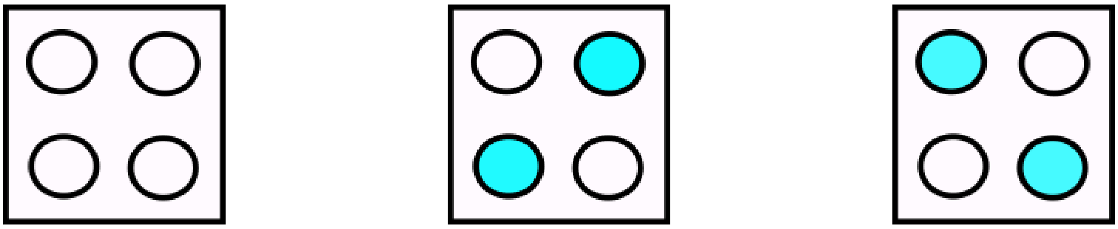
1.2 Beyond CMOS solutions: FCN and QCA

As previously anticipated, FCN is one of the solutions proposed in the international roadmap as a potential candidate to replace the current CMOS technology. Above all the exciting characteristics defining FCN, the more appealing one is the possibility of eliminating current flow through wiring, which is one of the leading causes of power dissipation in electronics nowadays. Moreover, this would allow arranging quasi-reversible operations from a thermodynamic point of view [7], [8] .
FCN has been physically demonstrated with different implementations, such as the metallic [9] and magnetic [10]. In general, there is the possibility to obtain tiny devices that can be operated at room temperature [11]. Concerning the molecular implementation of the FCN paradigm, it relies on Quantum-dot Cellular Automata (QCA) structures. In

this solution, the idea is to have quantum dots as the low potential regions where the charge is more stable and more likely to stay.
This chapter introduces the basic principles of the QCA paradigm and then the realization in the molecular field.

1.2.1 QCA introduction

To properly understand how the field coupling paradigm works, it is essential to introduce the most common way in which it can be implemented [9], i.e., Quantum-dot Cellular Automata. The primary cell is represented as a square in which four dots are present, reported in figure 1.2.



(a) Basic cell QCA

(b) Logic '0'

(c) Logic '1'

Fig. 1.2: Basic cell schematic representation for QCA paradigm

Above are reported three important figures to understand the basic concepts behind the QCA implementation. Figure 1.2a reports the basic scheme of a single cell. The four white circles are the low potential regions where the electrons can be confined. The charges move into those dots according to external influences in terms of electric interaction. In particular, in a QCA cell, there are just two stable configurations where the charge can localize due to the need to minimize Coulomb repulsion [12]. These are reported in figures 1.2b and 1.2c.
Both these charge distributions can be associated with corresponding binary information arbitrarily chosen. Given this brief introduction, it is important to understand how information propagation occurs. For this purpose, look at figure 1.3.

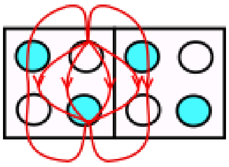


Fig. 1.3: Schematic propagation in QCA technology

In figure 1.3, it is reported the basic schematic of the propagation mechanism, in this case, is considered just the influence derived from one molecule on the others. The red lines across the dots are the electric field lines derived from the charge distribution between the two dots. According to the rules of electrostatic, the lines are moving out from the positive pole and entering the negative one. The voltage derived from this will have an opposite sign following equation 1.1.

$$E = -\frac{\Delta V}{\Delta s} \tag{1.1}$$

The minus sign indicates that E points toward decreasing potential. The electric field lines will influence the charge distribution on the other quantum dots in the circuit. Look, for instance, at the first two dots on the right-hand cell. In this case, the electric field deriving from the cell on the left forces the electrons to move in the opposite direction: the binary information can be propagated through the structure, as reported in figure 1.4. As already said, this configuration provides the most negligible value for Coulomb repulsion within the electrons [8]. As soon as the first cell switches its content, the whole wire will follow the same behavior, providing the propagation of binary values.

Notice that this explanation is just a massive simplification of the actual situation, in which all the electric field lines deriving from all the dots have to be considered.

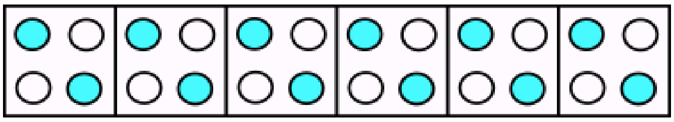


Fig. 1.4: QCA wire composed by six cells

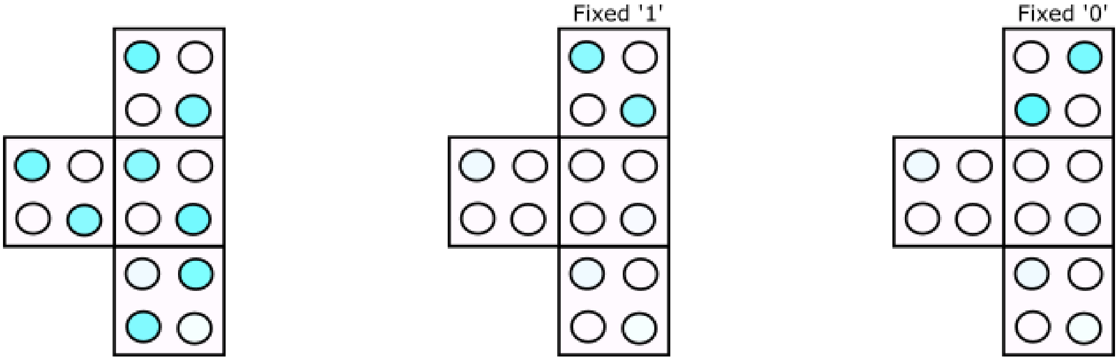
As a last remark for this introduction, consider the explanation about the electric field lines and the coupling within neighboring dots. From the physical explanation, it is more clear how this technique avoids energy dissipation as much as possible: there is no current flowing through the structure but charges moving in different quantum dots. In this way, the energy dissipated is in the order of $10^{-20}J$ [13], at least two orders of magnitude lower than the energy dissipated during a transistor switching.

QCA implementation of logic functions

Every technology aiming to replace CMOS must present the possibility of building all the elements allowing to reproduce the same blocks on which standard electronics are built, such as the more common basic gates and logic functions.

In particular, OR gates, AND gates, Majority Voters, and inverters must be present and working with this paradigm.

This section presents all the basic gates to deeply understand how the QCA paradigm can cope with the requests of digital logic design. In general, it is possible to say that a proper circuit layout is needed to implement a given digital circuit with QCA and FCN paradigms. Indeed, according to the cells specific positions, a different working behavior can be obtained. First, it is essential to introduce the majority voter [12], whose schematic structure is reported in figure 1.5a.



(a) MV gate (b) OR gate (c) AND gate

Fig. 1.5: Majority Voter, AND, OR gates layout

The circuit consists of three input cells placed on three different sides of a central one. The charge configuration of the central cell will be equal to the most present at its interfaces, as reported in figure 1.5a. There are two logical '1' inputs and just one logical '0'. According to the need to minimize the Coulomb repulsion, charges distribute to provide a logic '1' value. Notice that this structure is highly relevant for this thesis since it will become the primary block of the neural layouts proposed in the following. About that, it is crucial to introduce the concept of polarization of a QCA cell, which is the primary solution to measure the effective distribution of charges among the dots. This concept will be recalled in several parts of this work, and the formula can be appreciated in 1.2.

$$P = \frac{Q1 + Q3 - Q2 - Q4}{Q1 + Q2 + Q3 + Q4} \tag{1.2}$$

Therefore, it is possible to have a numerical evaluation of the charge distribution among the four dots of the cell.
Notice that, in the previously shown example related to the majority voter, since each cell is influencing the others, there will be different kinds of influences. Indeed, the color representation reported in the figure is just a simplified sketch of the situation. All these considerations will be analyzed in more detail in the following parts of this thesis.
As previously anticipated, it is possible to realize other basic logic gates from the majority voter structure. In figure 1.5b and 1.5c are reported the layouts for the OR and AND gates, respectively. In figure 1.5b is reported the OR gate layout, as said, it is the same of a majority voter except for a fixed input representing a logic '1'. On the other end, figure 1.5c reports the AND gate layout. Given the presence of the fixed input at '0', to obtain a logical '1', the other two inputs must be configured as logic ones. According to the usual AND cell truth table, the output will be a '0' value in all the other cases. FCN paradigm implemented with QCA technology has also proven reliable for more complex circuit solutions [14]. An example is the inverter, reported in figure 1.6.

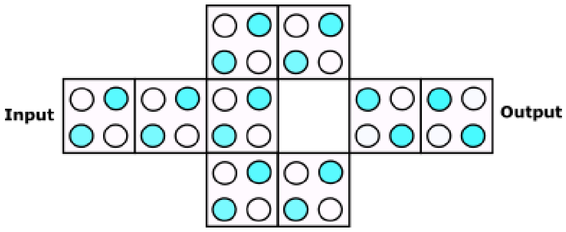


Fig. 1.6: Inverter layout

As it is possible to see in figure 1.6, the input and output configurations are opposite in the structure. The circuit takes advantage of the diagonal coupling within the cells, providing the inversion of the propagating information.
Usually, in CMOS technology, the inverter is one of the primary and more straightforward realizable gates. In QCA, the inverter is already not that straightforward to be implemented.
Another structure example can be, for instance, the XOR gate, analyzed in [14], which is even more complicated and whose structure has to be subjected to proper adjustments to provide a correct propagation.

Clock structure in QCA

Among the general rules governing the QCA paradigm in terms of functionalities, one of the most important is the division of the circuit in regions controlled by different clock signals. Those regions are activated only when the corresponding clock signal has an active value, allowing the electrons to move through the dots according to the incoming input voltage.
The reasons for which the insertion of the clock signal is essential are several. Some of them are listed below, together with some hints about the layout design:

- Avoid the loss of information for long propagation. Indeed, for too long wires, the information risks being lost. Indeed, as soon as the clock signal is released, the charges tend to move randomly in one of the two stable configurations if the correct information doesn't arrive in a concise amount of time.

- Provide an unidirectional propagation of the information

- Allow sending more information with a certain time separation, introducing in this way a sort of pipelining procedure

- with proper modification of the cell structure, the clock signal will force the charge into intermediate positions to make the switch easier from one dot to the other. Therefore, the cell must show six dots instead of four. The two additional ones will be used as charge containers as soon as the clock for that cell is non-active. The schematic of this new cell is reported in figure 1.7. The cell is in a reset state if the charge is confined in the two other dots. It is important to remark that these two intermediate regions are not stable positions for the charge.

- A correct clock region design has to avoid possible propagation clashes; this is a fundamental concept. Thinking again of a long wire made by QCA cells, it is reasonable to believe that as soon as the corresponding clock region goes active, all the cells in that area are ready to switch their content. However, the charge situation is not stable: the cells will tend to change to one of the two stable configurations as soon as possible. If the information meant to be propagated has not arrived, it might be lost in the propagation. A simplified example is reported in figure 1.8.

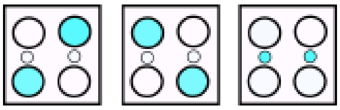


Fig. 1.7: Six dots cell schematic. Depending on the value of the clock field the charge can place itself according to the reported schematic

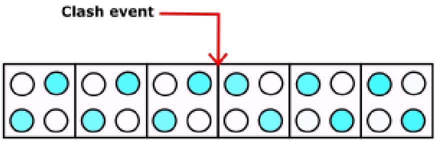


Fig. 1.8: Clash event schematic

It is, therefore, interesting to understand how to provide the clock to the circuit.
A single clock signal shows two flat regions and two transition regions. The reason for the smooth edges is double. First of all derives from technological limitations, so the impossibility to physically realize a perfect square wave. On the other hand, a smooth transition from negative to positive values reduces the power consumption [13] and contributes to avoiding possible metastable conditions during the propagation [15].
The general representation of the clock signal is reported in figure 1.9.

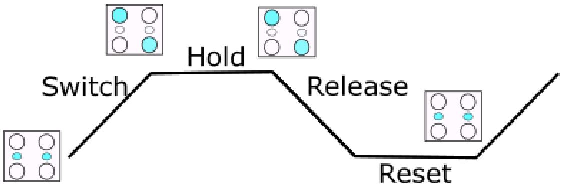


Fig. 1.9: Clock signal phases

Along with the signal's different time steps, the QCA cell behaves differently. As reported in the figure, there is the switch phase, the hold phase, the release phase, and the reset phase. The charge is gradually pushed towards the two dots in the switch phase

with the increasing electric field deriving from the clock.
The charge distribution is maintained at its maximum value in the hold phase when the clock signal is active. After the hold phase, the electric field from the clock reverses its sign gradually, and the charges are stably stored in the additional dots during the reset state. It is important to remark that this last condition is not a stable one for the charges from an energetic point of view. For this reason, they tend to move out of that situation as soon as possible to minimize Coulomb's repulsion.
This is the starting point to obtain correct adiabatic propagation; this consists of having the clock signals, one for each clock region, partially superimposed in time. This way, the information propagation is maintained stably in one direction, and the charge configuration is not lost. In the opposite case, charges tend to configure randomly in one of the two stable configurations, as previously explained.
A graphical representation of this situation is reported in figure 1.10. Notice how the signals are superimposed one with the other. As a ground rule, the falling edge of a first clock signal corresponds to the activation of the following. In general, the maximum and minimum levels of the clock field should be such that in the reset state is equal to $-2\frac{V}{nm}$ and in the hold state $+2\frac{V}{nm}$.

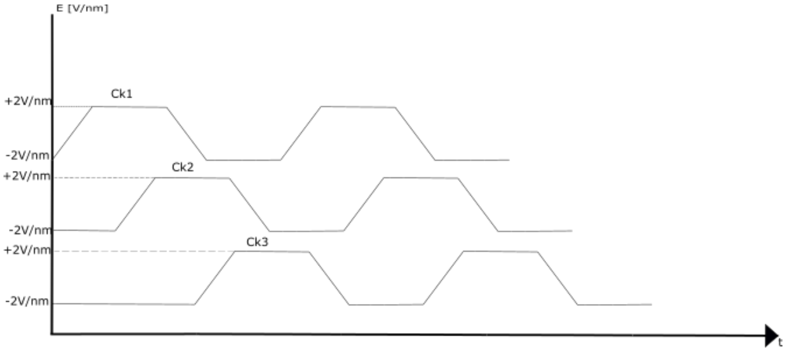


Fig. 1.10: Adiabatic propagation

So, it is interesting to look at the actual propagation expected when dealing with the schematic representation reported in figure 1.11. Imagine having three clock regions driven by the clock signals presented before. The information propagation across the cells is adiabatic since each clock zone is stably active while the following one is activating, allowing the correct information propagation through the wire. As soon as the first region is in a reset state, the input driver can change its configuration to propagate a new binary value at the following step. This way, it is possible to implement a pipelining procedure.

In figure 1.11 it is just reported graphically the same situation described in words in the previous paragraph. Notice how the information propagates along the wire. Initially, only the first clock region is subjected to the input driver influence. Then, the second signal moves out from the reset phase, and the cells polarize according to the neighboring charge distribution. The second region has to influence the following one in the third step. It is essential to remark that the second region must be stable and sufficiently long to maintain the information until the last zone is entirely active. At the same time, the

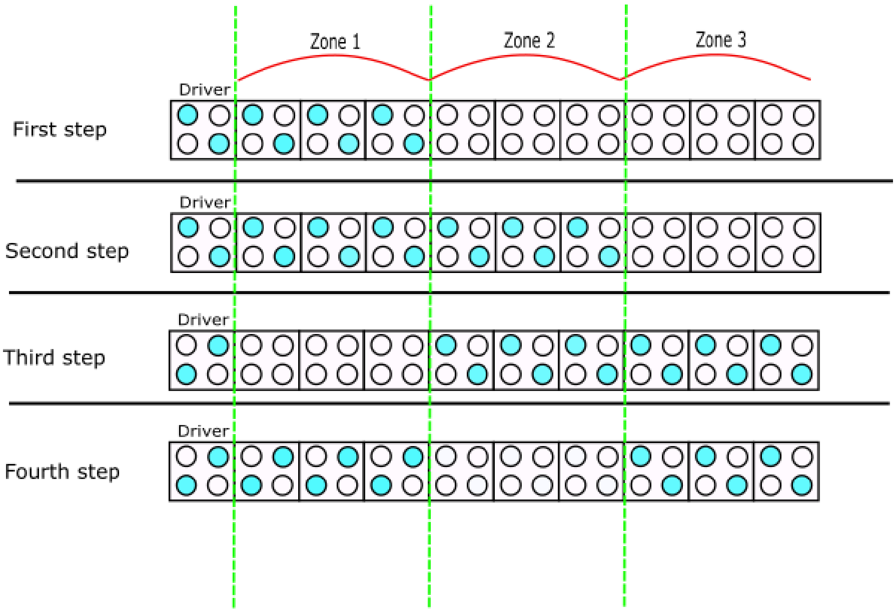


Fig. 1.11: Adiabatic propagation through the wire

first region deactivates, and the driver can eventually change its value. The situation goes this way for all the time needed to end the propagation, with logical '0' and '1' successively propagated through the structure.
As a final remark, as reported in [14], all the circuits previously presented can be subjected to clock signals and clock division. In all cases, the information propagation improves using the clock signals.

1.3 Molecular FCN

The previous sections introduced the rules that must be respected and understood to deal with the FCN paradigm implemented with the QCA solution. From a practical implementation point of view, research is moving in different directions, leading to different technological solutions showing positive and negative aspects. Previously was already mentioned the ongoing work on magnetic interactions [10] [16]. Other solutions rely on GaAs/AlGaAs heterostructures, given the possibility of creating quantum wells with those. These have the problem of impossibility to lateral branching [17], making them useless if the interest is building logic functions. Notice that nowadays, the most used solution is the magnetic one.
In general, to develop a possible technology for QCA, the structure must be one of the schematic cells reported so far in terms of characteristics. So for each cell, there is the need to have six dots, and moving charges should be present.
Another exciting possibility is the molecular one. Specific and ad-hoc synthesized molecules are used and properly arranged through self-assembly to develop the QCA cell.

Molecules can represent a real breakthrough solution for FCN implementation thanks to moving charges within the structure. Indeed, it is possible to state that molecules can contribute to implementing extremely dense devices due to their nanometric size. The power consumption is minimal, and molecules have been proven reliable when working at room temperature [18]. From a technological point of view, molecules can also provide some advantages in the fabrication process. Indeed, if the goal is to design devices at nanometric sizes, there is the need to involve complex fabrication processes for lithography to achieve such small nodes. Working with molecules implies taking advantage of their self-assembly [19] property, which can be considered rather simple to other techniques. Moreover, there is the possibility to work at extremely high frequencies, as reported in [20].

1.3.1 Molecules for molecular FCN

In the previous paragraph, some interesting characteristics of molecules have also been cited from a technological point of view. The main problem in this aspect is finding proper candidates to act as molecules for FCN implementations. In general, the molecules have to show the following characteristics:

- they must present moving charges, passing from one dot to the other through tunneling
- charges have to be stored in redox centers
- there is the need for a third dot in which the charges are localized in the reset state
- must be sensitive to an external electric field
- the electric field can be applied either in the vertical direction (clock) and longitudinal one (driving electric field)
- molecules have to be deposited through self-assembly on an underlying substrate

In the course of the still ongoing research in this field, have been proposed several candidate molecules. For instance, a possible single-molecule solution is analyzed in [21], in which the authors propose a single tetra-ruthenium complex to work as the entire QCA cell, having four dots in which the charge can be placed.

In general, the proposed molecules divide in the one presenting only three among the six needed dots and others that can be used alone due to their four dots plus a fifth keeping the charge in the reset state. Notice that it is not simple to manage all the technological requirements. For instance, the diallyl-butane is not a viable candidate given the impossibility of being linked to an underlying substrate [22].

Among all these possible solutions, the molecule that is going to be used and deeply analyzed in this work is the bis-ferrocene, whose structure is reported in figure 1.12. This is one of the most studied and characterized molecules in literature [22]. As can be seen in the figure, the molecule is composed of two ferrocene groups working as charge containers during the active state of the cell, a carbazole group hosts the charges in the reset state, and finally, a thiol group. The function of this last element is structural since it allows the link between the molecule and the gold substrate below [23].

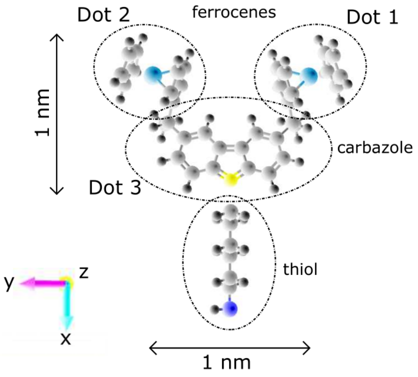


Fig. 1.12: Bis-ferrocene molecular structure [23]

The bis-ferrocene molecule has been ad-hoc synthesized for molecular FCN applications, as reported in [19] and [24]. From figure 1.12, it is possible to identify the three dots that can store the charge according to the involved influence. Dot1 and Dot2 are the low potential regions in which the charge is stored to encode the logic state. Notice that a single molecule alone is not stable and does not allow to have all the six dots previously presented for QCA implementation. For this reason, the standard cell is made up of two bis-ferrocene molecules that, together, form the needed six-dots cell.

1.3.2 Model and simulations

In FCN solutions, it is fundamental to find strategies to correctly evaluate how propagation occurs; the analysis should be done in the most computationally efficient way. For what concerns the analysis of molecules in quantum chemistry are well-established ab-initio simulations [25]. These can be used to study the interactions present within a single molecule or between different molecules. Therefore, molecules can be characterized rigorously. However, the main problem is that the computational weight is very high; since it purely relies on quantum mechanic analysis, the calculations are expensive in terms of time and computation. This solution is completely unusable if the goal is to deal with complex molecular QCA circuits. Therefore, it is imperative to define a set of algorithms and figures of merit that well represents the different situations taking into account the more reliable representation of the molecule behavior.

In the course of the work the adopted methodology relied on SCERPA algorithm [26] [14] [27].

As for any electronic device, the basic building block, i.e., the molecule, has to be characterized. This first step is crucial since all the definitions of charge distributions are analyzed and will be used to model the molecule behavior as soon as it is inserted into a complex circuit.

To characterize the molecule, the previously mentioned ab-initio calculations are adopted.

In particular, the density functional theory is involved [28]. The characterization of the molecule is provided by observing and saving the molecule response when subjected to an external driving field. Look at figure 1.13a to have a model representation of this analysis. The figure reports a typical situation that can be analyzed whenever it is needed to characterize a new molecule. The molecule under test is subjected to an external electric field derived either from a point charge or another molecule with a specific charge distribution. The electric field is directly proportional to the voltage difference among the dots and, therefore, the charge distribution on the driving molecule. Notice that the simulation of just two coupled molecules is already quite expensive from a computational point of view. For this reason, it is essential to limit this type of analysis to such complexity. At the simulation level, the distance between the molecules must be chosen accordingly to the actual working scenario, i.e., the lattice constant of the material onto which the self-assembly monolayer will be performed.
Another critical type of analysis is the one related to the presence of an external clock field [29], in that case, the electric field is applied along the vertical direction, providing the charge to be distributed in the carbazole or between the two ferrocene dots. The schematic of this analysis is reported in figure 1.13b. It is possible to appreciate a very simplified schematic of the performed evaluation. A longitudinal electric field generated by a point charge simulates the presence of an external clock field. In particular, it is supposed to be such that the electrons move to the two upper dots. Then, the application of the electric field deriving from another molecule or, more in general, from a second electric field perpendicular to the first one will move the charges to one dot or the other.

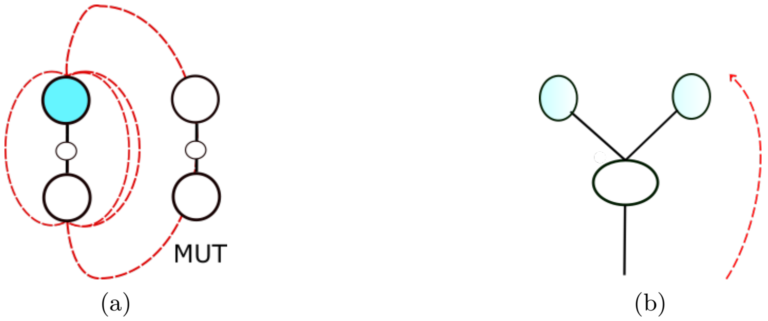


Fig. 1.13: Schematic representation of ab-initio simulations. Left: top view of molecule to molecule interaction. Right: external voltage influence analysis

The results of the ab-initio calculations must then be processed. At this point, some new figures of merit have to be defined to read the behavior of a certain molecule easily and comparable to others.
There are several figures of merit that can be defined and described [29], some of them are listed here below:

- aggregated charge (AC): the overall charge in each dot computed as the sum of all the atomic charges. This FOM is also involved in the definition of other essential characterizations of the molecular behavior

- Electric field generated at the receiver: this is the parameter of a particular molecule under test that works as a driver subjecting another molecule at a certain distance. With the previously defined aggregated charge is relatively easy to compute the electric field generated at a given distance using the formulas of standard electrostatics.

- V_{in}- AC Transcharacteristics (VACT): this is probably one of the most important figures of merit related to the subject. As already explained in this introductory part, the molecular FCN paradigm core is that the molecules charges distribute according to the input electric field. Using the formulas of electrostatics is straightforward to find the input voltage to the subject molecule, described in 1.3.

$$V_{D,\mathrm{MUT}} = \int_{\gamma} \boldsymbol{E}_D \cdot d\boldsymbol{l} = V_D\left(\boldsymbol{r}_1^{\mathrm{MUT}}\right) - V_D\left(\boldsymbol{r}_2^{\mathrm{MUT}}\right) \tag{1.3}$$

E_D is the electric field generated by the charge distribution on the driver molecule and γ is the path depending on the relative position of the two dots. Another positive aspect of relying on very known and easy formulas is that it would be possible to analyze rotations and misalignments within molecules in detail without increasing the computational weight. At the same time, it is possible to evaluate the aggregated charges on the different dots within a certain input voltage range. Therefore, it is possible to define VACT curves, which describe the relation within the incoming electric field and the aggregated charge among the dots.
The VACT can be defined for different clock values, the one reported in figure 1.14 is related to the bis-ferrocene molecule subjected to Ck $= +2\frac{V}{nm}$.

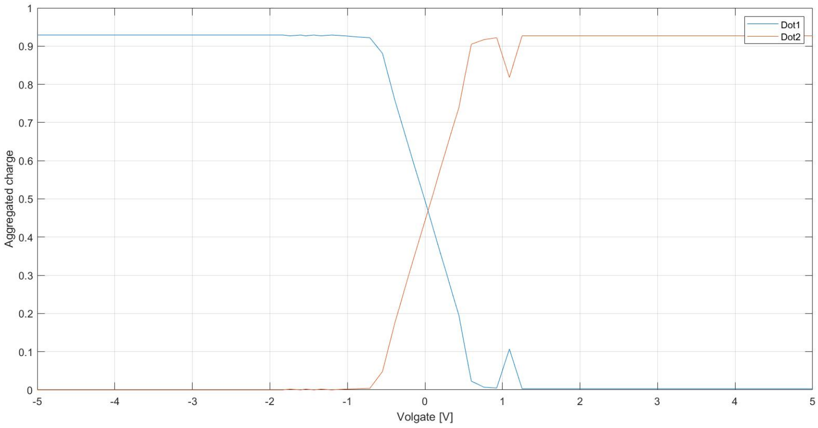


Fig. 1.14: VACT of the bis-ferrocene molecule

Before moving on, just a tiny remark is necessary. As it is possible to notice in figure 1.14, the two curves for dots one and two are not perfectly symmetrical and do not cross zero. This behavior is due to the not perfectly balanced and ordered structure of the molecule itself, and it will be taken into account several times in this work.

- From the aggregated charge characterization is also possible to derive the output voltage generated by the molecule, giving rise to a V_{in}/V_{out} curve from which it is possible to evaluate the gain of the molecule. Its value cannot be higher than one since the aggregated charge is always lower than one. In the best conditions, the gain value reaches the unitary value.

Self-Consistent ElectRostatic Potential Algorithm (SCERPA) introduction

After defining the main characteristics of each candidate molecule, it is possible to move on with the third and last stage of the analysis of a molecular FCN circuit. In this case, something more complex must be done since the system complexity increases. In particular, as noted previously, ab-initio evaluation cannot be considered a possible solution at this stage.
Think at this point to use bis-ferrocene molecules. The interaction among the molecules in the layout defines the aggregated charge distributions. SCERPA is a comprehensive algorithm that has been proven extremely useful in analyzing complex molecular QCA systems [27]. The central core of the algorithm is solving the self-consistent field loop present due to the interaction between the molecules and the surrounding ones. Imagine first a simple situation in which only two molecules are present: the charge distribution of the first one depends either on the input driver voltage to which it is subjected but also on the V_{out} of the second molecule. Indeed, the aggregate charge separation in the second molecule produces an electric field affecting the polarization of the first one.
The mechanism described is accurate for each molecule involved in a complex system. For this reason, the algorithm must solve a loop whose exit condition is to reach stable values of aggregated charges in all the molecules. Of course, notice that the mutual influence within molecules is valid provided that the two are not too far from each other. In that case, the electric fields are too weak to influence the other molecule; SCERPA considers this concept through the interaction radius parameter.
The working flow of SCERPA can be divided into three main steps [27], listed and commented here below:

- Initialization stage: the layout, the clock regions, and the driver voltages are defined as well as the biasing conditions to each molecule

- Interaction stage: the interaction within the molecules is taken into account, and the loop is solved. The algorithm refines the solution until it reaches a user-defined convergence condition.

- output stage: after evaluating the voltage contributions at each molecule, the algorithm, taking advantage of the VACTs of the involved molecules, can identify the final aggregated charges at all the dots, providing graphical and numerical representations of them on output files.

1.3.3 Technological implementation of molecular FCN

Regarding technological implementation, the literature does not offer too many hints. However, some important research related to the subject can be found in [30], [23] and [31].
This section offers a general idea about this kind of technology technical implementation and limitations. First, consider figure 1.15 which offers a schematic representation of the system physical structure. It is a simplified schematic of a bis-ferrocene wire deposited on a gold substrate. As already mentioned, the molecules have to be provided with a part whose aim is to attach to the below metal film systematically; for what concerns the bis-ferrocene molecule, the thiol group is the part devoted to this task. The technological solution used is the self-assembly monolayer technique. It allows ordered molecule deposition to be obtained, provided that the gold substrate has a constant lattice.

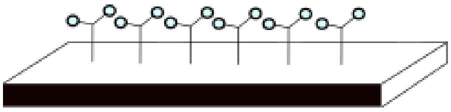


Fig. 1.15: Graphical simplified representation of a bis-ferrocene wire deposited on a gold substrate

Notice that the last consideration is not that straightforward since it is not easy to have deposition of the thin metal film with uniform geometry. The main risk is the possible presence of regions with different heights; in the end, there will be the presence of grains on the metal strip. The deposition then would not be ordered; molecules will show different intermolecular distances, one from the other. The risk is to stop propagating the information through the wire or, in general, through the molecular structure.
Another important aspect is the generation of the electric fields needed to make the system work. Two sources of the electric field have to be present, one in the longitudinal direction for the driver implementation and the other in the vertical to realize the clock field. The usually adopted solution consists of a trench structure: at the bottom of the trench is fabricated the gold film onto which the molecules are deposited. At the top of it are fabricated the electrodes used to generate the field lines that serve as input drivers and the clock field generation. In particular, the clock field is generated by applying a potential difference between the top electrodes and the gold nanowire at the bottom of the trench. The schematic representation of this trench technique is reported in figure 1.16.

It is possible to provide some important considerations from a technological point of view. The height of the trench must be chosen so that the electric field lines can effectively influence the charge distribution within the molecule. This idea must also be realized in the other direction, not to separate the two top electrodes too much and keep the electric field as effective as possible. So the parameters that must be controlled during the technological process are the height of the gold nanowire, the size of the molecule, the depth and width of the trench, and the parameters related to self-assembly realization. As a last remark, avoiding complete process variations in this structure is difficult. The

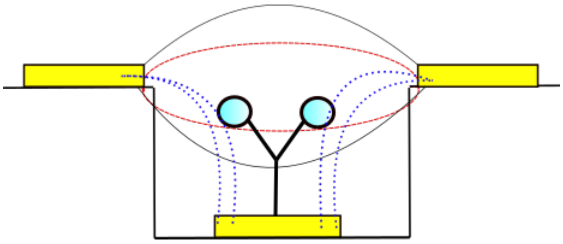


Fig. 1.16: Trench solution for the molecular FCN implementation

problem related to the presence of grains has already been mentioned; another important one concerns the impossibility of precisely realizing the clock regions division. In that situation, some problems arise:

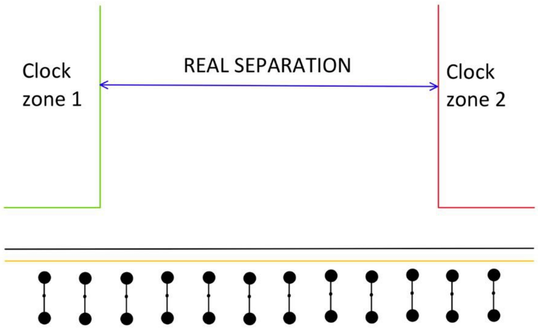


Fig. 1.17: Real separation of clock region in a physical implementation of FCN paradigm. Some molecules could not be directly subjected to a predefined clock region, being them in between of the electrodes providing the electric field lines

- the electric field lines coming from a particular clock zone, so from a specific pair of electrodes that subjects a portion of the circuit, could also affect other molecules. Actually, this is not a real problem. The electric field lines are in the direction orthogonal to the propagation, which will not be affected dramatically.

- the effect of the driver is not subjecting the first molecule of the circuit but a particular portion of it. However, this effect can be considered negligible if the information propagation moves in one direction only.

- A more severe problem is related to the limited precision of the technological solution through which the electrodes are fabricated, i.e., the resolution of the lithographic technique involved. The issue is having a certain number of molecules in between two clock regions, not directly subjected to one or the other, as reported in figure 1.17. Again, if the flow is unidirectional, the charge configuration of these molecules will be the correct one. The solution to this problem is a very well-controlled adiabatic propagation.

Chapter 2

Artificial neural networks: an overview

Artificial neural networks are one of the most promising and explored paradigms in machine learning implementations [32]. The number of different applications in which it is possible to find this kind of architectural solution is vast. These are pattern recognition, voice coding, writing recognition, face detection, and smart grids[33].

The main advantages related to the introduction of neural networks derive from the possibility of building them either in software or hardware and the flexibility associated with both these realizations. Considering the human brain as a reference starting point in technology can be proven to be a winning solution. First, the brain is a highly efficient machine, capable of achieving computation ratings in the range of 10^{17} FLOPS [34]. The basic idea behind artificial neural networks is to introduce in electronics or informatics the concepts at the base of the brain functionality. Of course, the purpose is not to reproduce the same behavior with similar computational characteristics, which would be simply unfeasible for nowadays technology.

This chapter will deal with the prior helpful knowledge for the proposed work. The analysis will be directed toward a hardware implementation of the artificial neural networks.

2.1 From human brain to hardware implementation

The human brain is formed by millions and millions of neurons coupled together. Each neuron elaborates the electrical signals coming at its input and, depending on those values, will spike, i.e., activate, or not. A spiking neuron generates a peak of potential, which is then propagated along the axon. The synapse is the input connection linking two or more nerve tissues. It is so the physical interconnection within two neurons. The main components of interest in the biological neuron [35] are listed here below:

- Nucleus/Cell body: this is the computation center for each neuron. The neuron body evaluates the weighted sum of the inputs connected to it through the synapses. If the resulting value is higher than the spiking threshold, the neuron activates, and the action potential propagates [36].

- Axon: this is the part of the neuron whose aim is to propagate the action potential. As in all interconnection wirings in digital electronics, the information derived from the neuron is not to be lost either by inversion or lossy propagation effects.

- Synapses: these are the interconnections within the input dendrites and the central body of the cell. In biology are usually labeled as chemical and electrical according to the working behavior.

Previously a fundamental concept was introduced related to the weighted sum of the inputs: it is how each neuron spikes or not. Different models of how a neuron spikes have been proposed over the years; among all, the Hodgkin-Huxley model [37] [38] is probably the most famous and adopted from an electronic perspective. The fact that the different inputs have different weights is significant because it increases the flexibility of the neuron itself. Moreover, the brain has high plasticity, meaning it can adjust according to different needs and external inputs. The synaptic plasticity translates in practice into the dynamic variation of the weights, allowing the brain to change and evolve in time.
All these concepts has to be translated into hardware.

2.1.1 Types of neural networks

It is possible to refer to a group of neurons connected as a neural network. According to how the information propagates in the algorithm, it is possible to identify two main categories: feed-forward networks [39] and recurrent networks. The first consists of propagation in only one direction; the network organizes into a set of mutually interconnected layers without feedback. Feed-forward networks can be considered the most accessible type of solution. On the other end, as the name suggests, the recurrent networks include cycles, allowing to the implementation of more sophisticated algorithms at the price of a more complex physical realization.
This work will deal mainly with feed-forward neural networks, whose schematic representation is reported in figure 2.1.

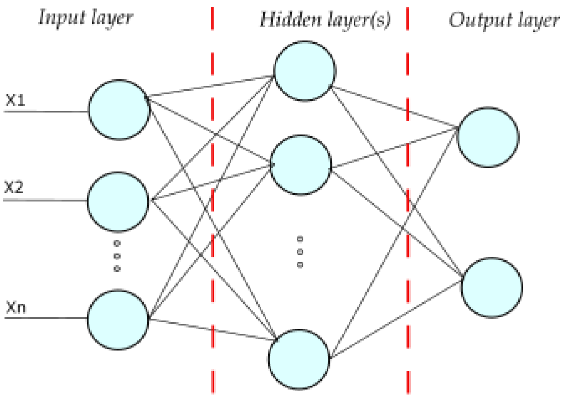


Fig. 2.1: Feed-forward neural network schematic

In general, in a feed-forward neural network, it is possible to provide a three-layer division: the input layer is the one in charge of taking the inputs, eventually multiplying them by a certain weight, and then distributing the results toward the inner layers of the structure. In the hidden layers, which can be one or more depending on the network complexity and task, the starting information is evaluated by all the neurons; at each neuron site, it is possible to provide different coefficients and operations. In the last stage, the output layer is present. There, the output results are computed.
It is important to detail the single neuron structure and analyze how it can be transposed into hardware. This is presented in figure 2.2. It is possible to notice the block representation that implements what was briefly exposed previously. The inputs are evaluated and multiplied by certain weights. Then, the single results are summed together. The output follows a certain activation function, which is the curve describing the spiking behavior of the neuron. If the input result is higher than the neuron threshold, the neuron can be considered active, and the spike occurs. Notice that the activation function is not unique; different implementations can have different activation curves according to the needs or the technology implementing the neuron. The mathematical equation used to model the single neuron behavior is reported in equation 2.1, where θ is an arbitrary bias that can be applied.

$$y_k = \sum_{i=1}^{N} b_{ik} \cdot x_i + \theta_k \tag{2.1}$$

In its simplicity, this formula will be fundamental in this thesis.

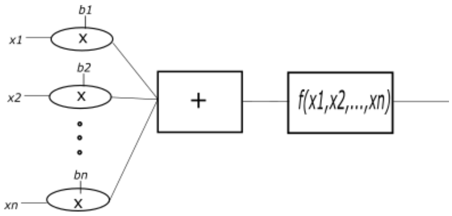
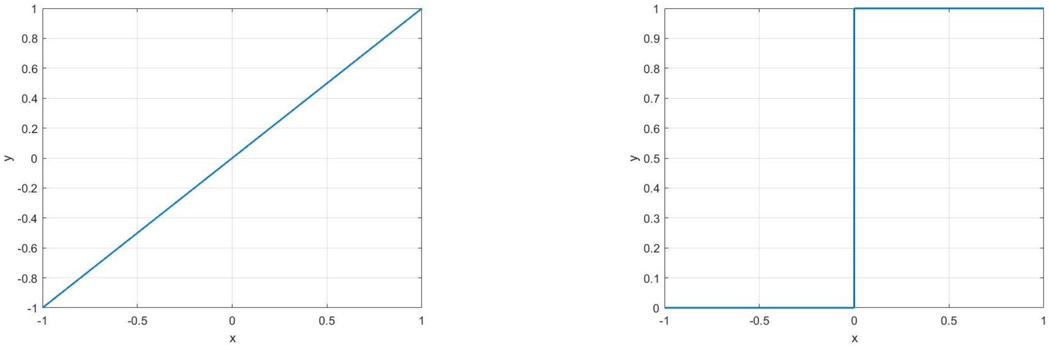


Fig. 2.2: Single neuron block scheme

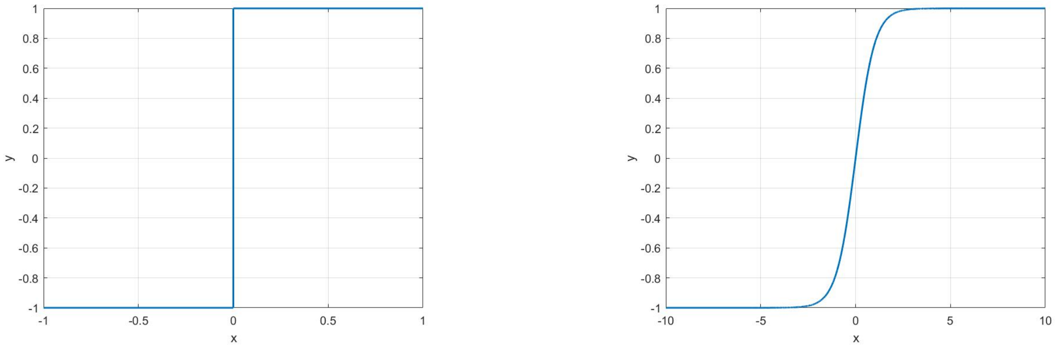
Different activation functions depend on the kind of equation that describes them. The need for a non-linear activation function in the neuron implementation derives from the fact that a linear curve is too simple to provide complex tasks, such as sophisticated mappings or information extraction from non-correlated input data. Therefore, an equation with a degree higher than one must be adopted. Moreover, to implement optimization strategies, the activation function has also to be differentiable [40]. Some examples are listed below and reported graphically in figures 2.3:

- Step function
- Hyperbolic tangent
- Sigmoid
- sign function
- exponential
- linear: better suited for linear regression models [40]
- leaky ReLu function [40]



(a) Linear activation function

(b) Heaviside activation function



(c) Sign activation function

(d) Hyperbolic tangent activation function

Fig. 2.3: Examples of activation functions

As already said, the chosen activation function has to meet the necessity of the system to be implemented. On the other hand, the activation function could be defined by the components present in the circuit for a hardware-implemented neural network.
In conclusion to this section, it is interesting to make a first and fundamental example of a realizable neuron: the perceptron [41]. It was the first-ever realized neuron, and will be central in the following discussions. It is often used as a binary classifier, given that its activation function is the Heaviside equation.

2.1.2 Learning mechanisms

In the previous section, was introduced the concept of synaptic plasticity. It is how the brain adapts to improve specific tasks solving capabilities. This concept translates into the need for an initial training or learning process in which the network adapts to solve the job in the best way possible to minimize errors. In the same way, artificial neural networks have to work to maximize performances. There are mainly two ways in which a generic neural network can be trained [42]:

- Supervised learning: in this technique, the network is trained with known inputs, and the results are compared with a scheme known as a-priori
- Unsupervised learning: in this case is the network itself that learns how to interpret a cluster of an unlabeled dataset, i.e., a set of data that the trainer does not know how to classify in principle

The learning step main goal is to ensure that all the neurons in the network have the correct weights so that the output results have a minimum error to the expected ones. In general, it is possible to say it is an iterative process. Concerning the training of a hardware neural network, there are mainly three possibilities [32]. The first one consists of off-chip learning. The system, in this case, is trained in a software environment, which provides the fastest learning possible. The main problem is that there is no indication of the actual and physical behavior of the network. Indeed, this is something testable only directly on the realized chip. To overcome this problem, it is possible to adopt on-chip learning, in which only the hardware is used. With this technique, the choice of the weights, besides being slower, can be considered more reliable. There is then a third possibility, which can be considered in-between the previous two. In this case, both the software and hardware are inserted into the training loop. The main advantage of this solution is the higher precision that can be associated with the weights, thanks to the presence of the software's computational capabilities.

2.1.3 Possible hardware implementations

There are different ways to realize neural networks physically. First, CMOS digital circuits can be adopted; the main advantage of this solution consists of the advanced and well-known fabrication processes related to CMOS [43]. Moreover, such a device can easily interface with external circuitry and standard electronics. In this category, the solution involving FPGA is worth mentioning as attractive given the flexibility of these devices [44].

In general, to provide a new hardware solution for ANN, the main effort must be put into the physical realization of its main features. These are the weights, that in CMOS can be implemented with latches [32], the threshold mechanism [45] and the activation function, which is related to the hardware realization of the neuron.

For what concerns the activation function, it is possible to say that it represents the most complex part to be realized, given its native non-linear behavior. Look-up tables have to be introduced [32] to achieve such a result.

A second possible physical realization relies on analog circuitry; despite being more complex to be realized on a large scale and having higher area consumption, this kind of solution is interesting from the efficiency point of view [46]. Other interesting aspects of analog ANN are the possibility of working in continuous mode and reaching stability through feedback. Solutions of this type are already available in the literature and have proven to be working [47], [48]. An exciting solution for the training methodology of analog ANN is reported in [49].

The main issue related to the analog solution is the proper update of the weights, particularly in an on-chip learning phase. Commonly resistors are used to represent weights. Since the main physical quantities in an analog circuit are voltages and currents, resistors may offer an easy way to implement the sum. Unfortunately, once a resistor is inserted into the circuit, it is impossible to modify its value during the training session. For this purpose, some new technologies have been introduced to overcome the problem.

The most promising is the memristor [50]. These devices can be thought of as two-port elements with variable resistance whose physical realization was performed for the first time in 2007 in HP laboratories [51]. Despite it can be obtained through a well-known fabrication process, such as the production of thin-film oxide, the interfacing with the CMOS circuitry is not that straightforward.

More complex devices adopting this technology are becoming useful on a commercial scale, such as the Resistive Random Access Memory (RRAM) [52], [53].

As a last possibility, it is worth mentioning hybrid technologies, where there is the presence of both the analog and digital parts, achieving the option of taking the main advantages of the two combined [54].

Chapter 3

FCN paradigm and neuromorphic computing

In recent years, hardware neural networks have been a real breakthrough in the technology environment. The application range they can be involved in is vast, ranging from robotics to biomedical and consumer electronics. Several examples of technologies involving neural networks were reported in the previous chapter. Therefore, it is crucial to find new possible solutions from a technological and architectural point of view to increase the efficiency and connectivity of hardware-implemented ANN.

This chapter gives a complete overview of the possibility of implementing ANN in a non-conventional technology such as molecular FCN. In this way, what was explained in the first two chapters is linked together to explore this possibility.

In recent years, such solutions have been explored; in particular, the primary ground steps were provided by E.P. Blair and S.Koziol in [55]. They proposed using quantum-dot cellular automata paradigms to implement simple neurons able to provide neuromorphic computation.

The suggested system has the following characteristics. To mimic the synaptic weight, it is possible to introduce partial clocking of the interfaces at the neuron inputs. The neuron spikes if the sum of all the input configurations, weighted by a non-complete activation of the clock zone, is higher than a certain threshold.

The threshold mechanism is fascinating since it involves introducing fixed cells whose influence could be either excitatory or inhibitory. The signals integration, i.e., the summation, takes advantage of the majority voting behavior, which is well known in QCA technologies. In figure 3.1 is reported a schematic representation of the solution proposed.

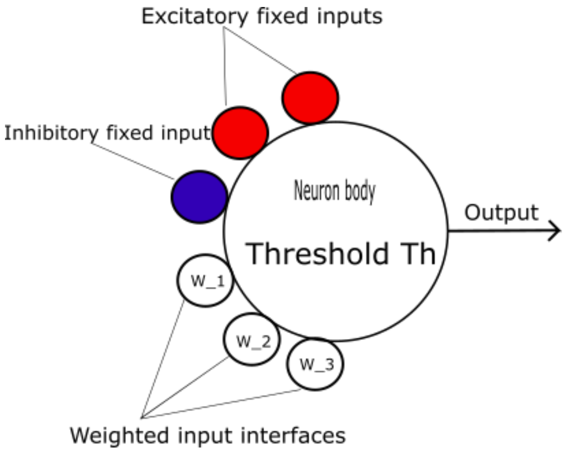


Fig. 3.1: First neuron schematic representation

It is exciting to adopt these basic ground rules and adapt them to the molecular case. The first advantage is the possibility of implementing the majority voting technique in the molecular solution for QCA, as commented in the first chapter of this work. One possible problem is related to the use of partial clocking. This technique implies that the last cells of the wire will not be completely polarized. This becomes a critical issue to be solved using molecules since the charge will not be stable in such a configuration, and information loss can occur.
For what concerns the threshold implementation, the method proposed by the authors in [55] is not that straightforward since it implies the addition of other interfaces at the input of the majority voter structure. Such solutions have not been analyzed in detail in the molecular field for quantum-dot cellular automata.
For this reason, the grounding rules exposed in this section can work only as a reference for proposing a new implementation of such neural networks.
A step forward in the analysis and realization of neuron-like structure using molecular FCN has been made in [56], and this is going to be the starting point of the study proposed in this thesis.
The basic structure of the single neuron cell is reported in figure 3.2.

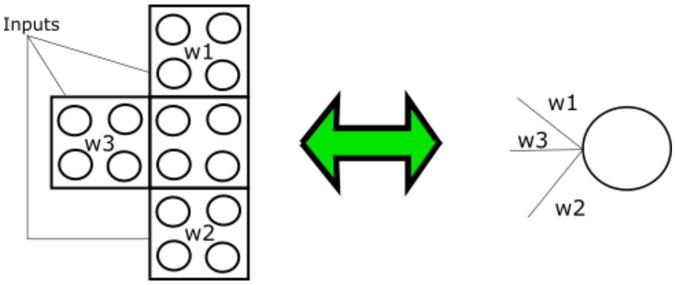


Fig. 3.2: Neuron cell schematic and usual representation comparison

It is helpful to highlight the essential concepts on which the analysis basis. Provided that the circuit layout underneath the molecular neural network is the well-known majority voter, the main characteristic of the system is the presence of more than one kind of molecule along the circuit. In this way, different molecules work as distinct interfaces to the MV, giving rise to varying values of weights. The majority voting procedure considers the different polarization levels at the inputs, and, according to those, the output is calculated.

A further exciting point is that for some combinations of input voltages and interfaces, the usual behavior of the majority voter can be reversed. Indeed, the less present configuration at the MV inputs can be reported on the output, provided that it is applied on the more robust interface, i.e., the one having the higher weight. It is a fundamental result that will be shown in the following.

The possibility of working with the majority voter layout has one major problem: up to now, only three inputs voting techniques have been analyzed in detail, meaning that only neurons with three inputs could be implemented with accuracy. It is a limitation that concerns the realization of multilayered neural networks. However, the majority voting allows to realize, with the molecular FCN paradigm, the equation reported in 2.1, i.e., it is possible to apply, with some tolerances, the simple neuron model to the molecular neuron layout.

Above in this section have been reported some hints related to the evaluation of the weights. As seen in the first chapter of this thesis, several figures of merit can be introduced to characterize a candidate molecule to work with the FCN paradigm. Among those, there is the P-V_{in} curve.

Different molecules can show different kinds of AC-V_{in} relationships, this depends on several characteristics of the molecule itself. For instance: the presence of counter-ion [19] [57], the molecule's nature, and the molecule's geometry. An important parameter that defines the different molecules involved is registered in [56] as α. It is the minimum input voltage for which the molecules saturate, i.e., show the maximum charge distribution among their dots. Different molecules have been ad-hoc synthesized in a simulation environment, with the same geometry but different α values, and for each of them, was evaluated the polarization curves induced on the central cell by the molecule under test influence. The weight of the cell made by the molecules under analysis is defined as the angular coefficient of this curve; this concept will be clearer later in this chapter. This way, all the molecules are classified as interfaces to the neuron central cell.

As the last point, it is worth mentioning the possible implementation of the threshold mechanism, proposed and analyzed in detail in [56]. It consists of applying an external voltage on the molecule at the output of the neuron cell. The resulting voltage must overcome a specific bias in output to the central cell. Another possible solution is to find molecules such that the deriving QCA cell has a certain tendency toward a positive or negative polarization.

Summing up these necessary considerations, it is possible to state that:

- the neuron cell is going to have the same layout of a majority voter gate
- the cells at the interface of the central one, i.e., the neuron body, will be made of ad-hoc synthesized molecules

- A certain weight characterizes each molecule; this is given by the angular coefficient of the induced P-V_{in} curve evaluated on the central cell.
- the equation used to characterize such a structure is the one already reported in the introduction to neural networks part (2.1)
- the threshold mechanism can be delivered with particular kind of molecules or with the application of an external voltage on the first molecule present after the neuron body

After this introductory part related to the fundamental reasoning behind the structure of a possible neuron made of molecules, it is necessary to provide an overview related to the organization of the research work done in this thesis.

3.1 Step 1: Analysis and characterization of the interfaces

Starting from the previously exposed ideas, it is possible to trace a series of steps followed to build up the final system: a working neural network based on the molecular FCN paradigm.
The first step consists of defining the molecules characteristics in terms of saturation voltage and analyzing their weights; this was done through the simulation layout reported in figure 3.3.

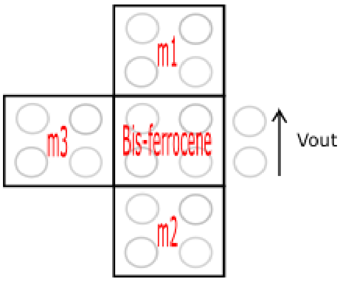


Fig. 3.3: Interface analysis schematic representation

The molecules in the central cell are bis-ferrocenes, while the output molecule is a dummy one, meaning that it will have zero charge distribution for any input voltage value. Thus it serves as a reference to take the output voltage. Interestingly, the other three interfaces are made of ad-hoc synthesized molecules, with the needed characteristics explained previously.
To evaluate one interface at a time, a voltage sweep is performed on the one of interest while the other two are left turned off. In SCERPA this behavior is relatively easy to implement: all the m1, m2, and m3 interfaces are driver molecules, showing a fixed charge distribution according to the input voltage they are subjected to.
In this way, it is possible to evaluate for each molecule on each interface the induced P-V_{in}

curve on the neuron central cell and the corresponding angular coefficient, expressed as reported in equation 3.1.

$$w = \frac{P}{V_i} \tag{3.1}$$

A second step to build up the whole system consists of reversing the bis-ferrocene positions with that of the molecules under analysis. The same kind of curves can be obtained and, consequently, their weights. Therefore, evaluating the bis-ferrocene molecule influence on the ad-hoc synthesized ones is possible.
The layout of this analysis is reported in figure 3.4.

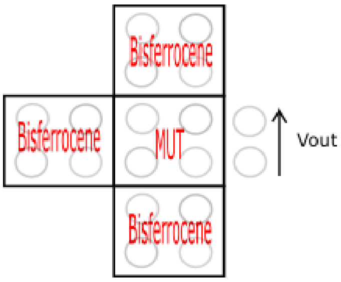


Fig. 3.4: Interface analysis layout: effect of bisferrocene on molecules under test

As a last consideration, it is noteworthy to consider the kind of molecules that will be taken into account. These are asymmetric molecules, meaning that their aggregated charge distribution ranges from 0 to 1 or, generally, from zero to a higher than one positive value. Moreover, the presence of the counter-ion is fundamental in keeping the molecule neutral for each input voltage. Thanks to that, possible crosstalk-related problems can be avoided, as shown in this work following.

3.1.1 Step 1.1: Analysis of a more complex interfacing mechanism

The next step needed to build up an exhaustive system analysis consists of connecting the two previous studies. In this case, bis-ferrocene drivers influence the ad-hoc synthesized molecules cells subjecting the central bis-ferrocenes. The main scope is to analyze the polarization of the bis-ferrocene cell subjected to different molecules driven by bis-ferrocene drivers. The importance of this analysis is to derive the differences from the previous case in which only the drivers were present at the interface, allowing to reach a more general and realistic point of view of the neuron behavior. Moreover, the layout that will be obtained is the one to use as the single neuron cell.
As done before, the polarization curve is the main instrument to evaluate the interface weight. The idea is to build up a table of weights and use them in a block way: knowing the cell present at a specific input, the importance of that interface will change, and the associated weight will go in linear combination with the input voltage coming from

the bis-ferrocene drivers. This layout, in the course of this thesis, will be referred to as *additional cell*. Figure 3.5 reports a graphical representation of what was just clarified.

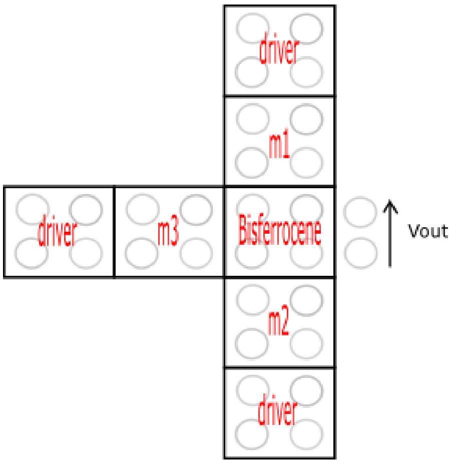


Fig. 3.5: Interface analysis layout: additional cell

Before building the system up, some numerical and graphical results will be given. In particular, it is interesting to understand how different combinations of interfaces and input voltages will behave if compared to the theoretical outcomes of the weighted linear combination of the input voltages, reported in equation 3.2.

$$Vout = w_{m1} \cdot V_{in1} + w_{m2} \cdot V_{in2} + w_{m3} \cdot V_{in3} \tag{3.2}$$

where w_{mX} and V_{inX} are the weights and input voltages at each interface (X = 1, 2, 3). Demonstrating this point is essential since this is the fundamental behavior of neural networks. If the error between the theoretical results and the obtained ones is not so high, it will be actual proof of the feasibility of the molecular neural network.

3.2 Step 2: Propagation analysis on a molecular wire

Molecular wires are needed to connect more than one neuron. Besides being analyzed deeply in literature [22], it is crucial to consider which is the best solution for the network application. In particular, some characteristics of the wire have to be studied and fixed to find the best way to transport the information derived from the computation in a first neuron to a second one. These are the length of the wire, i.e., the number of molecules present on the wire; the number of clock zones; the number of molecules needed in each clock zone; how the information propagates, i.e., analog or digital propagation.
This last point is significant since it will define the type of neural networks realizable with this technology. A fully digital application will imply the possibility of designing linear binary classifiers efficiently.

The layout proposed in figure 3.6 shows the basic schematic of a neuron to which an output wire is connected.

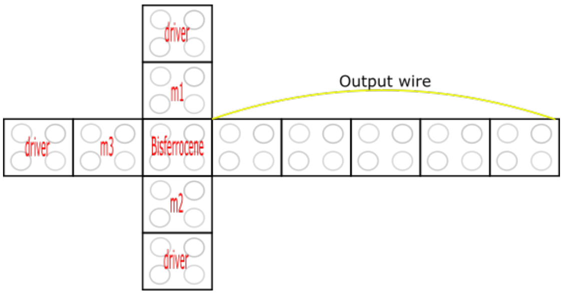


Fig. 3.6: Schematic layout of a neuron connected to the output wire. The number of cells in the wire is completely representative in this case

3.3 Step 3: first network structure and layout characterization

The next step in this work will be to build a neural network that considers all the reasonings provided about the necessary elements.
The first simulated is made of just three neurons in the input layer and another neuron in the output layer. The difficulty in providing such a solution is correctly propagating the information from different neurons synchronously. To do that, a specific and correct clock region distribution is required. That, as will be shown, moves away from the standard adiabatic one in some cases.
The necessity of an ad-hoc clock signal distribution derives from the presence of different molecules in the structure, which have no equal response to the application of the same electric field.
The structure that will be shown, built up step by step with exhaustive analysis for each interface, is the one documented in figure 3.7. To conclude, a feasibility study has been developed to compare software simulated neural networks with those built with the proposed set of molecules and weights. The outcome of this last analysis is a well-working neural classifier able to evaluate different input binary patterns correctly.

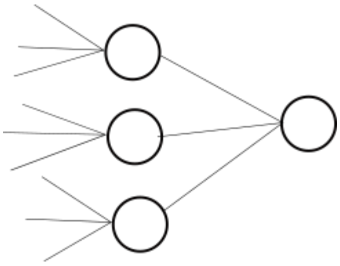


Fig. 3.7: First neural network schematic representation

Chapter 4

Interface analysis and weight evaluation

In the first part of the neuron analysis, it is necessary to define the primary building blocks that will be used in the study. In particular, the principal aspect treated in this chapter concerns the definition of different interfaces and their actual influence on the polarization of the central cell, i.e., on the neuron body. The interfaces have to be characterized from a physical point of view, defining the needed molecules to implement such a system. The influence of these molecules must then be evaluated numerically, and the weight of the interfaces extracted and compared with the other results.
Finally, has been also performed a parametric analysis of the weights according to intermolecular distance. The possibility of providing an FCN circuit made of different kinds of molecules is appealing also thanks to features inserted in SCERPA [56]: once the molecule is synthesized, so its VACT curves have been defined, they can be imported into SCERPA and used without the need of any further effort.
The definition of the VACT curves can be either extracted from computational analysis or built ad-hoc according to the scope. In this work, the second approach was preferred.

4.1 Definition of the molecules

As the first step in this analysis, as mentioned before, it is fundamental to define the type of molecules realizing the final neural network. According to the results provided in [56] the defined molecules are characterized by the following saturation voltages, i.e., α values: 1V, 1.5V, 2V, 2.5V and 3V. As it is possible to imagine, the higher the saturation voltage, the lower the charge distribution considering the same input.
The transcharacteristics are reported in figure 4.1, in which it is possible to verify the just provided observation. The reported results in figure 4.1 are examples of asymmetric transcharacteristics.

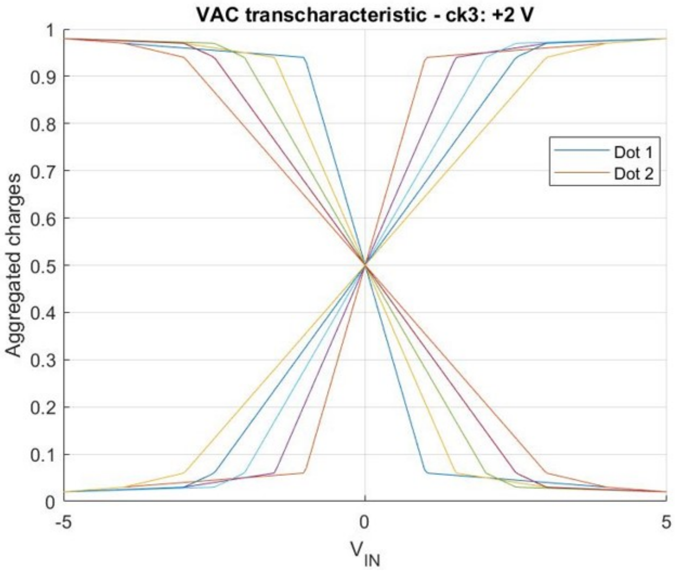


Fig. 4.1: VACTs for each of the proposed molecules

There is not much to say concerning the geometrical realization of the molecules. The most relevant thing is that they are built on the structure of the bis-ferrocene, having so two primary dots hosting the charge and a third one that is useful to keep the charge when the molecule is forced into a reset state.
The counter-ion has been proven a successful solution for the realization of molecular FCN solutions since it can solve or, at least, mitigate crosstalk. For this reason, there is a fourth dot whose charge value is such that the molecule is neutral for each input voltage. The global neutrality of a molecule is reached if, for each input voltage value, the sum of the charges contained in the molecule is 0.
As a possible graphical representation of what just said, look at figure 4.2, which reports a highlight on the VACT for the molecule characterized by $\alpha = 1V$. Notice the presence of the curve for the fourth dot having an aggregated charge value equal to -1 in the whole range of input voltages.
Notice that the aggregated charge values range from 0 to 1 for all the considered curves. Moreover, the curves cross the zero value and behave specularly for positive or negative input voltage values.
These last considerations are an idealization of the working behavior of the proposed molecules. It is important to remember that both of them are not true in the case of the bis-ferrocene molecule, which, as seen in the first chapter of this thesis, has an aggregated charge characteristic that does not cross zero.
As a last thing it is important to highlight that the curves present in figures 4.1 and 4.2 are both related to a clock value equal to $+2\ \frac{V}{nm}$. It is the more interesting but not the only one. In figure 4.3 is reported the aggregated charge distribution plot for a clock value equal to -2 $\frac{V}{nm}$. Notice that the difference in the α value is not of interest in this case since the charge can be at a maximum of 0.07 in the two upper dots; so, in practice, absent.

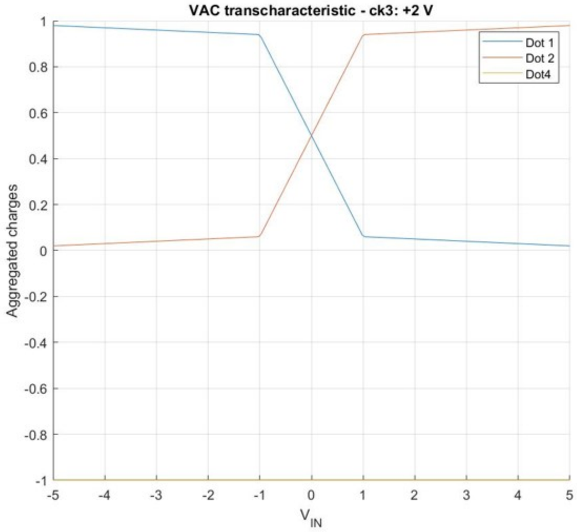


Fig. 4.2: VACT for $\alpha = 1V$ molecule with presence of counter-ion on dot4

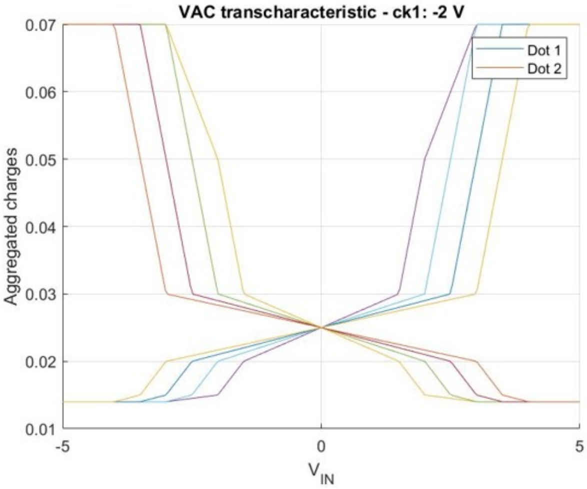


Fig. 4.3: VACTs obtained for CK = -2 $\frac{V}{nm}$

4.2 Analysis of the interfaces and weights extraction

Once the molecules are defined and characterized, the weight coefficients must be evaluated; these are useful for implementing the final neural network.
The weights are estimated starting from the polarization on a central cell made of bisferrocenes through a sweep on the input voltages on a driver cell made by the molecules previously defined. Figure 4.4 shows the basic layout structure.

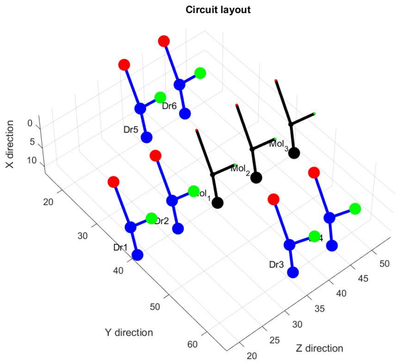


Fig. 4.4: Starting layout used for the weights extraction. This layout is going to be called *drivers only*

The elements present in the circuit are the following ones: the central cell, on which the polarization values are going to be evaluated, is made of bis-ferrocene molecules; the drivers are turned on one at a time with a sweep in input voltages that ranges from -2V up to 2V, in this way it is possible to extract the polarization curve; the last molecule, present at the output of the central cell, is a dummy molecule whose aim is to give a reference for the voltage evaluation out of the neuron body.
The polarization of the central cell will follow, as can be predicted from the basic theory of molecular FCN, the same charge distribution of the active driver.
In figure 4.5 is reported a snapshot representing the situation in which the m3 driver is active and made of $\alpha = 1V$ molecules, while the other two interfaces are off. The intermolecular distance along the y-axis equals 0.9nm, while the one along the z-axis is 2nm. The input voltage on the active driver is 0.5V. Notice how the charge distribution in the driver is not complete since made of molecules that saturate for voltages equal to or higher than 1V.

After this first introductory part, it is possible to move on and evaluate the weight coefficients. These must be computed for each molecule on each interface. The graphs are reported in the set of figures 4.6. All of them report the curves for all the three interfaces: dr3 is what was previously defined as *m3*, dr2 is *m2* interface, and dr1 corresponds to *m1*. Notice that the curves related to *m2* and *m1* are perfectly superimposed; this indicates the symmetric properties of these two interfaces. All the voltage values are normalized to the maximum V_{MAX} that can subject a molecule when inserted into an FCN circuit. It is possible to demonstrate that this value is 1V [56]. Some important considerations can also be provided regarding the curves. First, as already anticipated, the higher the α value, the lower the absolute polarization in the central cell. This means that the influence of the corresponding driver molecules is low.
To evaluate the weights, the angular coefficients must be computed. Since the curves are not perfectly linear in the whole range of input voltages, the coefficient is considered in the most linear region. For what concerns the value at which the central cell saturates,

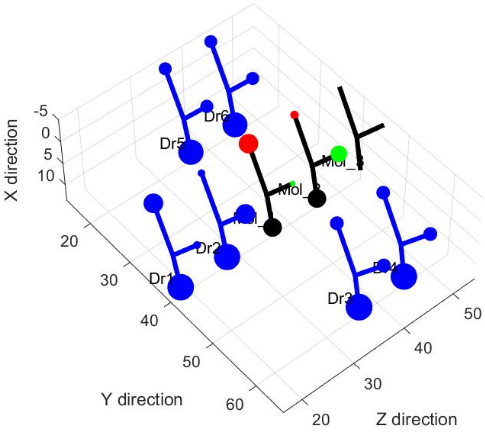
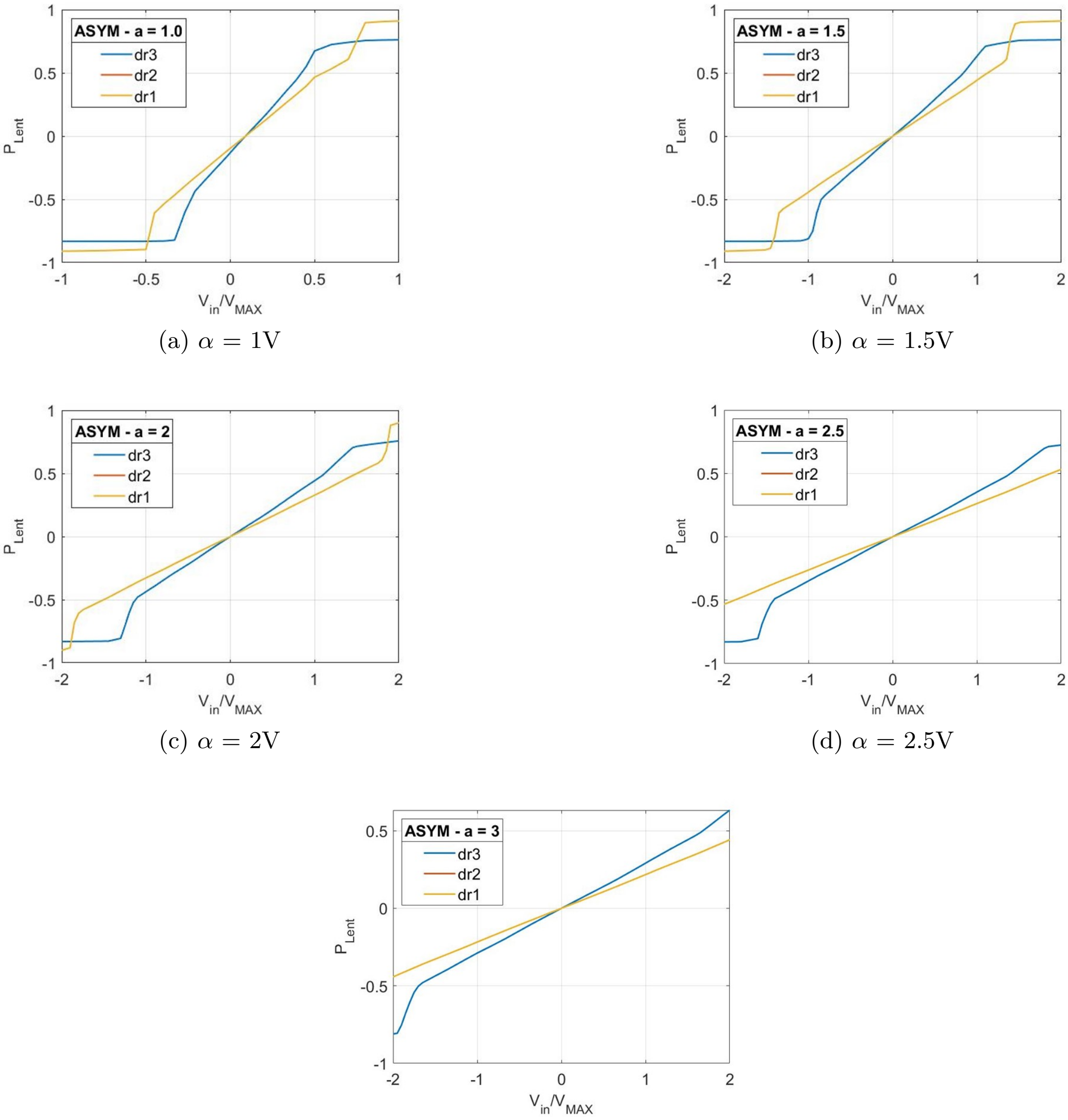


Fig. 4.5: Starting layout with active driver and charge distribution on the central cell. d = 0.9nm, $V_{in} = 0.5V$

it is clear that there is also dependency on the saturation voltage of the active interface molecules. In figure 4.6a, with 1V molecules, the saturation of the central cell is reached for $V_{in} \simeq \pm 0.5V$. The higher the saturation voltage of the molecule, the higher the input voltage needed to saturate the central cell. In the extreme case of $\alpha = 3V$, the saturation cannot be reached in the range $V_{in} = \pm 1V$.
Looking at the different curves, it is important to highlight that the m3 interface influences more than the other two. This behavior is due to the geometry of the chosen layout, implying a stronger influence of the electric field and, therefore, a higher polarization of the central molecules. As a final analysis of the polarization curves, the weights can be extracted. As already anticipated, the weights for m1 and m2 will be perfectly equal due to symmetry reasons. All the values are listed in table 4.1.

α	m1	m2	m3
1V	0.676	0.676	0.918
1.5V	0.45	0.45	0.622
2V	0.332	0.332	0.464
2.5V	0.266	0.266	0.37
3V	0.22	0.22	0.295

Table 4.1: Weights values: molecule X on bis-ferrocene



(a) α = 1V

(b) α = 1.5V

(c) α = 2V

(d) α = 2.5V

(e) α = 3V

Fig. 4.6: Polarization curves for the molecules under analysis

A further important thing to highlight is that with this technology, all the weight coefficients will be lower than 1. The gain of a QCA cell implemented with molecules cannot be higher than one; thus, the interface weight will always be between 0 and 1. This behavior is easily explainable given the intrinsic nature of propagation in molecular FCN: the voltage at the output of a molecule can be, at maximum, equal to the input one in case of saturation.

4.2.1 Voltage analysis of the *drivers only* layout

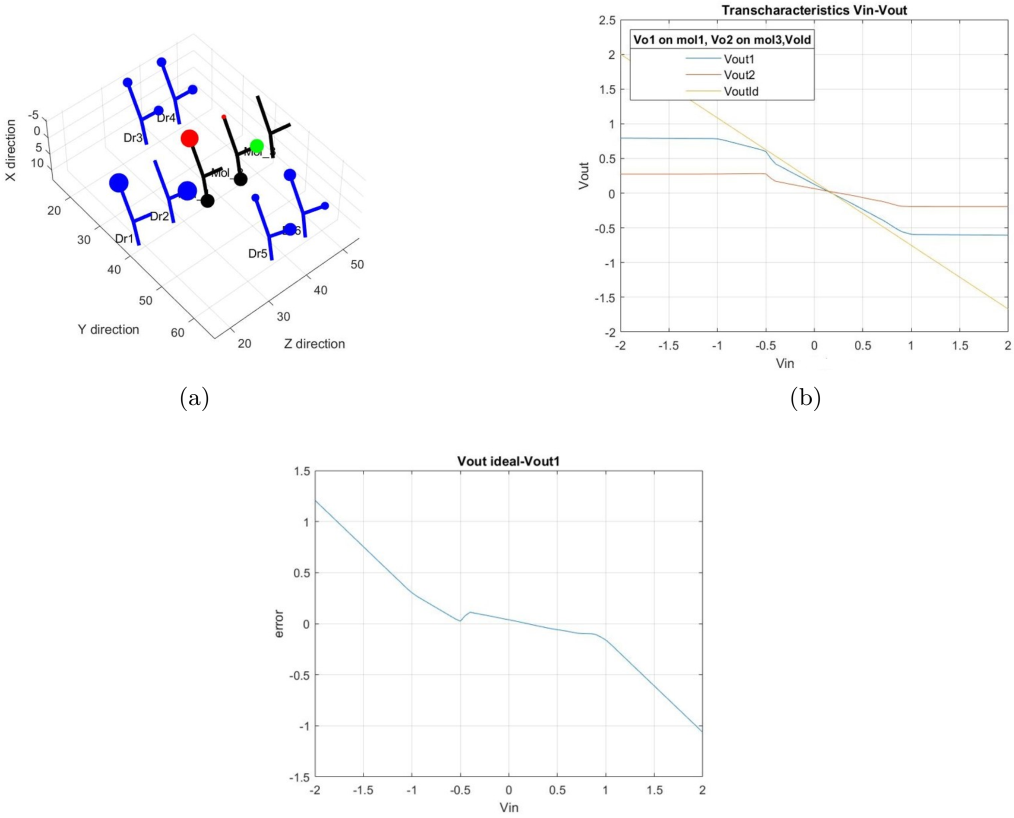
With the weights just calculated, it is interesting to make some first trials applying to the molecular structure the equation 3.2.
To accomplish this task, the procedure was the following: one of the three interfaces is subjected to a sweep in the input voltage values while fixed voltages influence the other two. The weight formula is evaluated in its ideal form on the whole voltage range. In this way, it was possible to understand whether or not the model works and find some critical regions in which the structure cannot correctly operate as a neuron of an ANN.
To compare the obtained results with those calculated with the ideal formula, it is possible to consider the voltage at the output of the left molecule of the central cell. In the proposed graph, the dummy molecule voltage is also present. It is helpful to have an idea of the voltage that will influence the first molecules of the output wire, which will be positioned in the next steps in the same position as the dummy molecule.
The first example is characterized by:

- **Interface m3**: input voltage sweep from -2V to 2V. $\alpha = 1$V
- **Interface m2**: $V_{in_2} = 0.5$V, $\alpha = 2V$
- **Interface m1**: $V_{in_1} = 0$V, $\alpha = 2V$. This interface is going to be non influent in this first example.

The results are reported in figure 4.7.

From this first simulation, some interesting analyses can be provided. Looking at the graph reported in figure 4.7b, it is possible to notice that in the region in which the output voltage on the central cell has a linear variation, the difference with the ideal curve obtained through the application of equation 3.2 is relatively small. This linear region falls approximately within -0.5V and 1V. For lower or higher values the voltage on the molecule saturates due to the saturation of the charge distribution within the dots. In figure 4.7c is reported the error curve; this shows minimal values in the linear region and then increases in magnitude at the extreme of the input voltage range.
The error increases because the adopted model does not consider the molecules saturation, meaning that the voltage prediction works well until the charge distribution is not maximized.
Another point is that the obtained voltage curve is not symmetric to zero but shifted towards positive values. This behavior is due to the presence of the fixed voltage provided by interface m2, which is a positive one. Therefore it enhances the polarization of the central cell when the interface m3 has the same configuration and reduces it when the charge distributions are opposite within the two drivers.
As a second example consider the following one:

- **Interface m3**: V_{in_3} =0.7V, $\alpha = 1.5V$
- **Interface m2**: voltage sweep from -2V up to 2V. $\alpha = 1V$
- **Interface m1**: V_{in_1} = -0.8V, $\alpha = 2.5V$



(a)

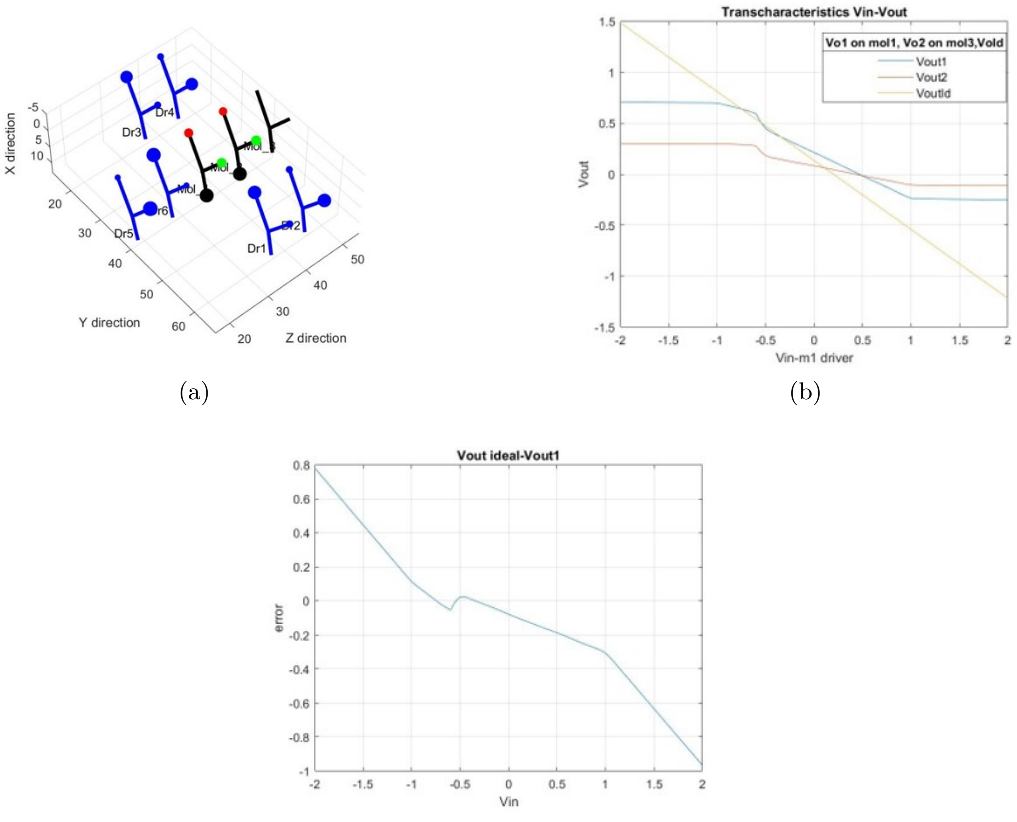
(b)

(c)

Fig. 4.7: Simulation results. 4.7a represents a snapshot of layout and the polarization on the involved molecules. 4.7b shows the output voltage curves. 4.7c reports the error curve: difference between the ideal output voltage and the one obtained on the first molecule of the cell

The simulation results are reported in figures 4.8. In this case, it is possible to notice an error higher than in the previous simulation but still relatively small in the same input voltage range. Therefore, once again, when the central cell is working in its linear region, V_{out} tends to follow quite well its ideal behavior.

The interesting point is that the shift toward positive values is even more evident than in the previous example; the output voltage curve is completely asymmetric to $V_{out} = 0V$. Notice in figure 4.8a that, besides two interfaces configured as logical ‘1’, the central cell has the charge not distributed. It means that the influence of m3, which is quite strong in this case, is such that it counteracts the contributions of the other two. An actual inversion of the majority voter behavior is not present but the result strongly opens to the possibility.



(a)

(b)

(c)

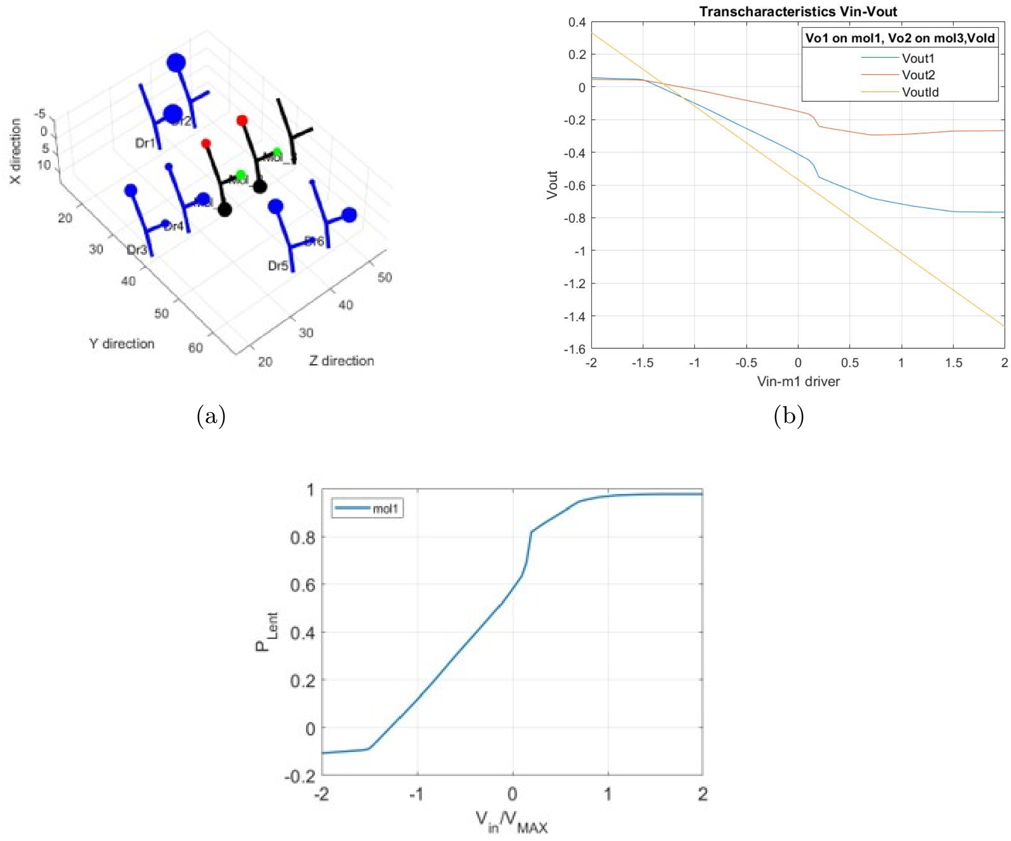
Fig. 4.8: Simulation results. m3: $\alpha = 1.5V$ $V_{in_3} = 0.7V$, m2 = sweep, m1: $\alpha = 2.5V$ $V_{in_1} = -0.8V$

At last, consider the following situation:

- **Interface m3**: V_{in_3} = -0.8V, $\alpha = 3V$
- **Interface m2**: V_{in_2} = -1V, $\alpha = 2V$
- **Interface m1**: Input voltage sweep. $\alpha = 1.5V$

The presented one is an extreme situation involving molecules with high saturation voltages. However, the results are useful to highlight the point; these are reported in figures 4.9. In 4.9b, it is clear how the output voltage shifts toward negative values. The maximum positive value that this combination of voltages and interface molecules provides is close to 0V. It is also interesting to notice the polarization curve of the central cell, noticing how it is almost completely on the positive side of the y-axis. This means that one of the two logical configurations is prevailing upon the other for this circuit. However, although two interfaces have the same configuration when m1 has the opposite

one, the output voltage will be slightly positive. Therefore, it is also possible to state that the voltage inversion will be stronger by using molecules with lower α on interface m1. This can be considered as the first example of inversion of the usual working behavior of the majority voter. In the following of this work, other examples will be reported.
In conclusion, the main result that can be highlighted is the possibility of predicting with a small error the behavior of the central cell of the neuron whenever the combination of interfaces and input voltages avoids cell saturation. Once the charge distribution is maximum in the cell, the mathematical equation used to describe the single neuron cannot properly track the behavior, which is restricted to the molecular implementation and not native to neural networks. In the proposed examples, the errors in the useful region are quite small, but besides the detailed result analysis, it is interesting to notice how the sign of the predicted voltage is always the correct one. This last consideration will be extremely useful and analyzed in-depth in the following.



(a)

(b)

(c)

Fig. 4.9: Simulation results. m3: $\alpha = 3V$ $V_{in_3} = -0.8V$, m2: $\alpha = 2V$, $V_{in_2} = -1V$, m1: sweep

4.3 Toward a complete structure: the *additional cell* layout

After evaluating the single interfaces and the different influences that the new simulated molecules have on the polarization of the central cell, it is essential to move on and complicate the structure to reach more realistic modeling of the circuit layout. We will refer to this second solution as *additional cell* case. The interfaces will not be used as drivers but are directly subjected to the influence of driver cells made by bis-ferrocenes and transport the information to the neuron body. Therefore, the input voltages from the drivers are weighted by the interfaces. The central cell computes its configuration according to the weighted sum of the inputs. The basic structure of this solution is the one reported in figure 3.5.
Before analyzing this complete structure, it is crucial to understand the type of influence that the bis-ferrocene drivers are going to have on the interfaces.

4.3.1 Effect of bis-ferrocene drivers on MUT cells

This paragraph explains the analysis previously mentioned. The basic layout is reported in figure 4.10. As done in the first analysis, have been evaluated first the polarization curves and, consequently, the weight coefficients of such interfacing. Starting from the m3 interface, i.e., the longitudinal to the MUT cell, the results are reported in figure 4.11, in which is documented the case for $\alpha = 1V$. It is interesting to look at figure 4.11b, which reports the polarization curve evaluated on the cell made of $\alpha = 1V$ molecules. The main difference from the circuit with the bis-ferrocene, evaluated in the previous section, is the lower polarization value reached in correspondence to the input voltage range boundaries. These are equal to 0.3 and -0.3. Also, in figure 4.11a a not complete polarization of the cell under analysis can be seen. This behavior is clearly due to the high saturation voltage that such molecules show, which prevents a complete charge separation among the upper dots of the molecules. At this point, it is straightforward to say that the higher the V_{sat} value, the lower the maximum values reached by the polarization curve.
The results for interface m3 are reported in figure 4.12.
As correctly predicted, the higher the saturation voltage, the lower the charge distribution. The polarization is almost absent for the $\alpha = 3V$ case.
At this point m3 interface is completely described, and the analysis can move toward the other two. The polarization curves are reported in figure 4.13. Again, the layout is symmetric along the z-axis.

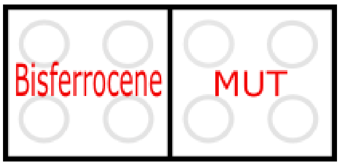
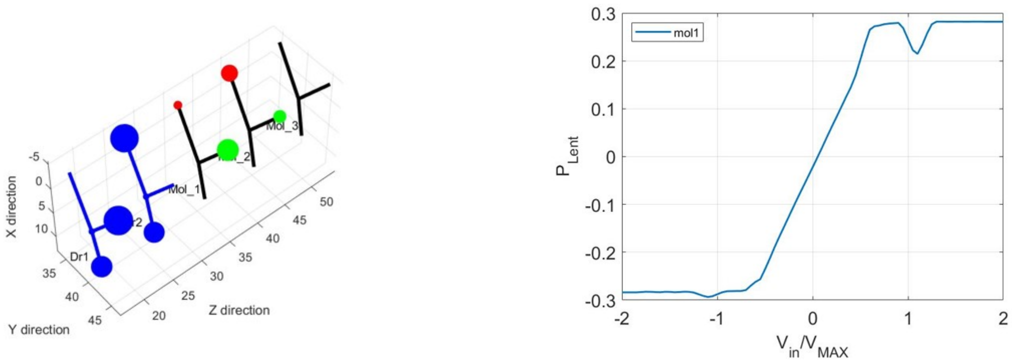


Fig. 4.10: Adopted layout for the analysis of the influence of bis-ferrocene on MUT cells



(a)

(b)

Fig. 4.11: Figure a: circuit layout. Figure b: polarization curve for $\alpha = 1V$ on m3

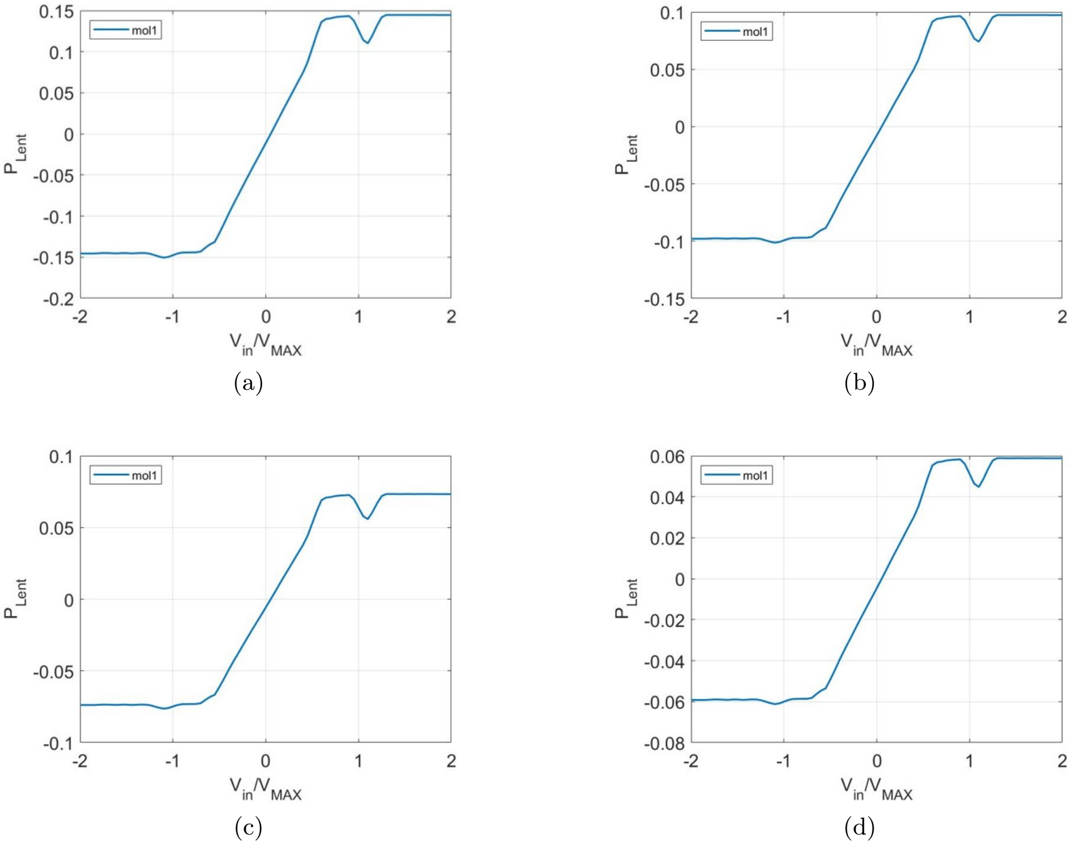
In general, the polarization values found are lower than those for interface m3. It is due to the geometrical realization of the circuit, which has a larger molecule separation along the vertical direction. All the results follow the expectations. The extracted weight coefficients are reported in table 4.2.

α	m1	m2	m3
1V	0.297	0.297	0.395
1.5V	0.152	0.152	0.209
2V	0.102	0.102	0.144
2.5V	0.0772	0.0772	0.107
3V	0.062	0.062	0.085

Table 4.2: Weights values: bis-ferrocene on molecule X

4.3.2 ***Additional cell*** layout and the problem of crosstalk

Adding a further cell in the structure is a step toward the final analysis of the basic structure of a single neuron. The structure consists of a bis-ferrocene cell working as a driver; the molecules directly couple with the interface cell, built with a pair of the ad-hoc synthesized molecules. Then, in a second clock region, there is the central cell of the neuron, in which the computation will take place. In figure 4.14 are present the two snapshots of the simulation involving $\alpha = 1V$ molecules. From figure 4.14 it is possible to notice the influence that a specific interface has on the polarization of the neuron body made with bis-ferrocenes. The polarization level is not high either at the interface or the central cell. The interesting point that can be analyzed here is the back-propagation from the central cell to the interface molecules. This phenomenon will increase the polarization



(a)

(b)

(c)

(d)

Fig. 4.12: Figure a: $\alpha = 1.5V$. Figure b: $\alpha = 2V$. Figure c: $\alpha = 2.5V$. Figure d: $\alpha = 3V$. Interface m3

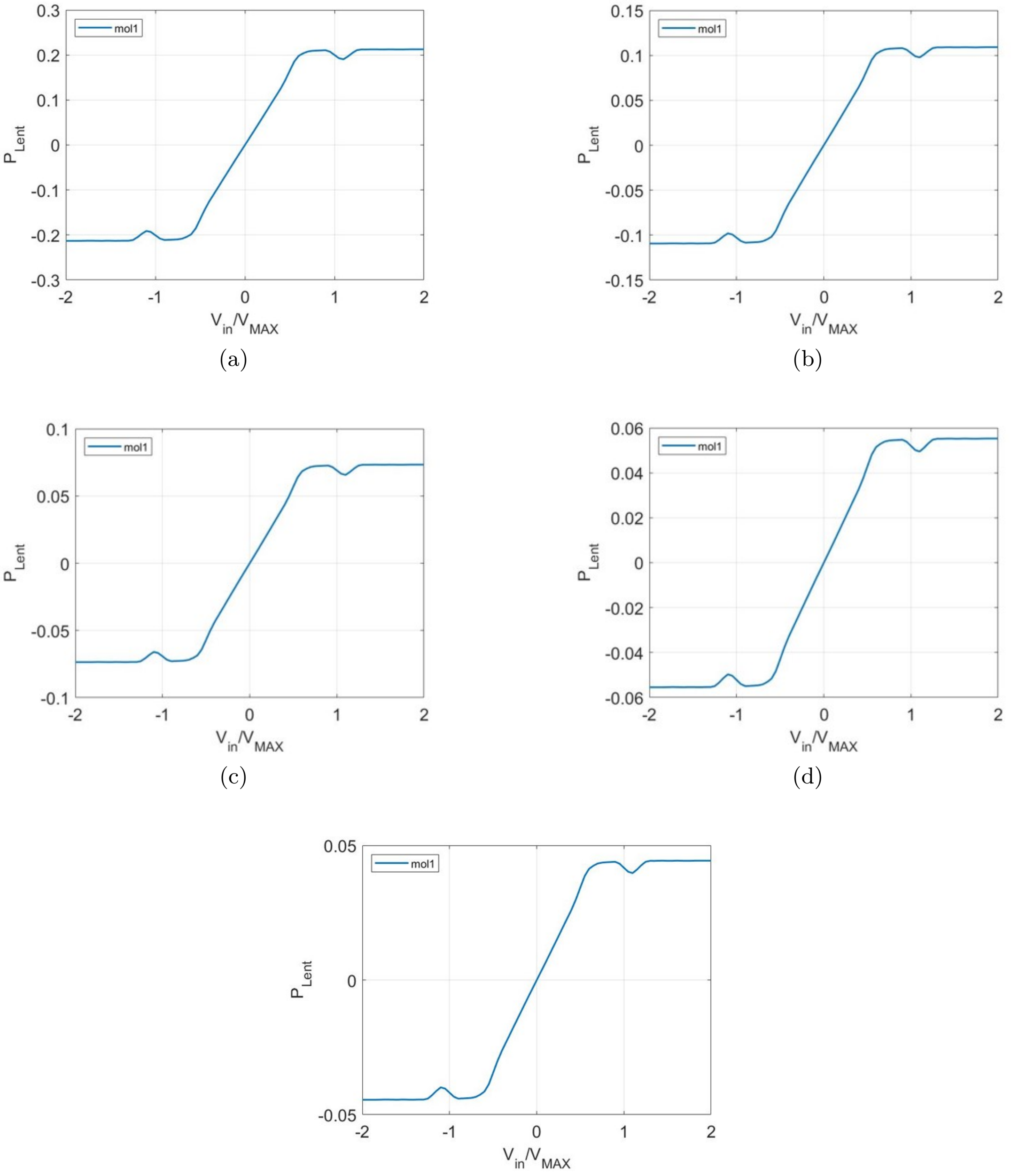
at the interface side when the second clock region goes active. Another aspect that must be considered is the border effect on the last molecule.
In figure 4.15 are plotted the polarization curves in different moments of the propagation.

To properly read these graphs, consider the following notation:

- **Interface before**: this is the polarization curve of the interface when the central cell, i.e., the second clock region, is still turned off. Interestingly, the same was found in the previous analysis of bis-ferrocene on molecule X interaction.

- **Interface after**: this is the curve representing the polarization of the interface cell when the second clock region is active. This curve is equal to the one defined as "interface(CK2)" in the other graph. The back-propagation effect is evident in the increased polarization.

- **Central**: this last polarization curve refers to the central cell behavior.



(a)

(b)

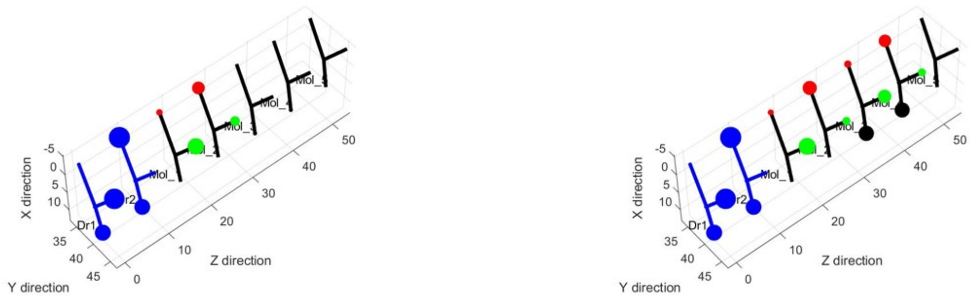
(c)

(d)

(e)

Fig. 4.13: Figure a: $\alpha = 1V$. Figure b: $\alpha = 1.5V$. Figure c: $\alpha = 2V$. Figure d: $\alpha = 2.5V$. Figure d: $\alpha = 3V$. Interface m1

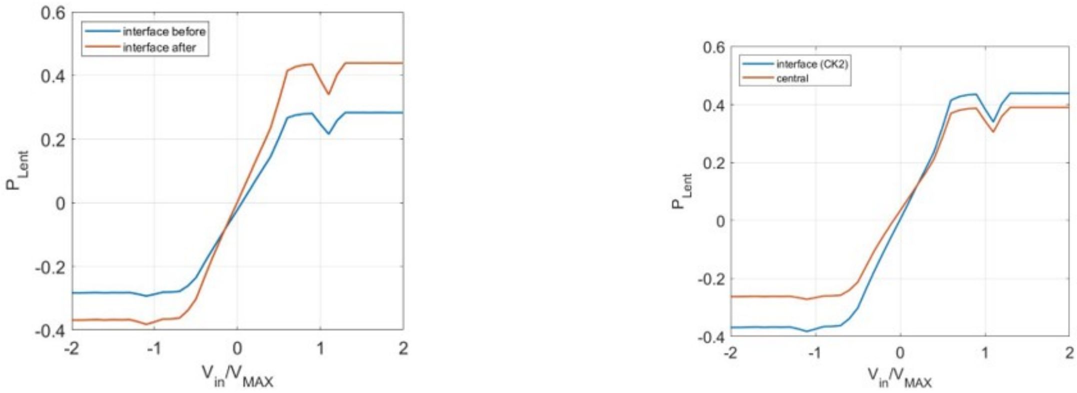
Looking at figure 4.15b, it is possible to notice how the polarization curve of the bis-ferrocene central cell ranges from around -0.2 up to 0.4; these values are lower than the



(a)

(b)

Fig. 4.14: Interface m3. Information propagation with interface $\alpha = 1V$

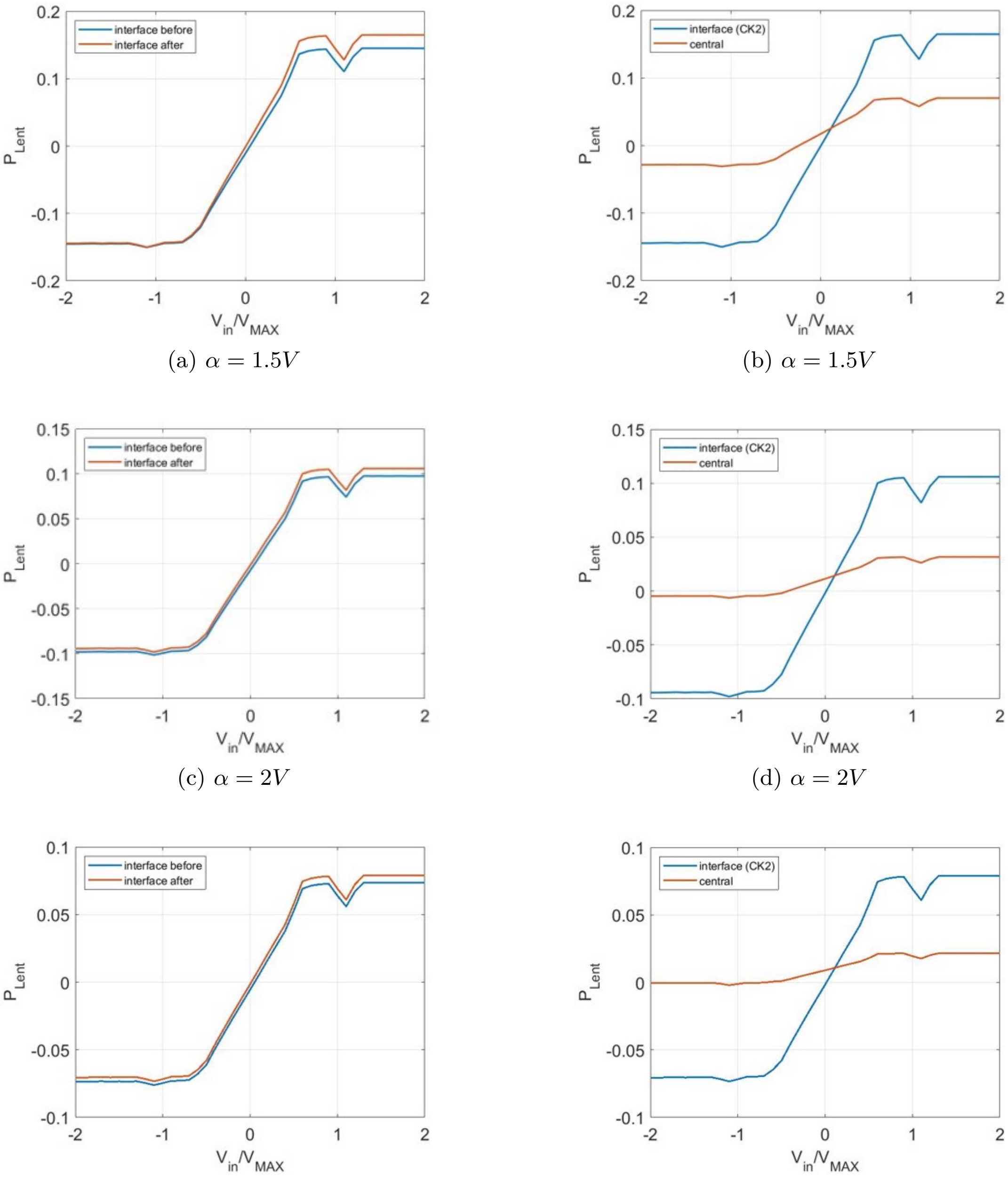


(a)

(b)

Fig. 4.15: Interface m3. Polarization with interface $\alpha = 1V$.

ones found in the *drivers only* analysis. The reason for that is the real presence of the interface molecules in this situation, which provides a decrease in the polarization and lower output voltages due to their high saturation values. For this reason, to properly understand the effect of a given molecule at the interface of the complete structure, it is needed to re-evaluate the weights and, therefore, consider the polarization curves of the central cell in different situations. This procedure must be done for all the molecules on all the interfaces. The values obtained are the ones involved in the voltage analysis that will be provided later on in the study of the neural network. As expected, the new coefficient values are lower than in the previous cases.
The plots present in figure 4.16 report the polarization curves for all the molecules under test placed on interface m3.



(a) $\alpha = 1.5V$

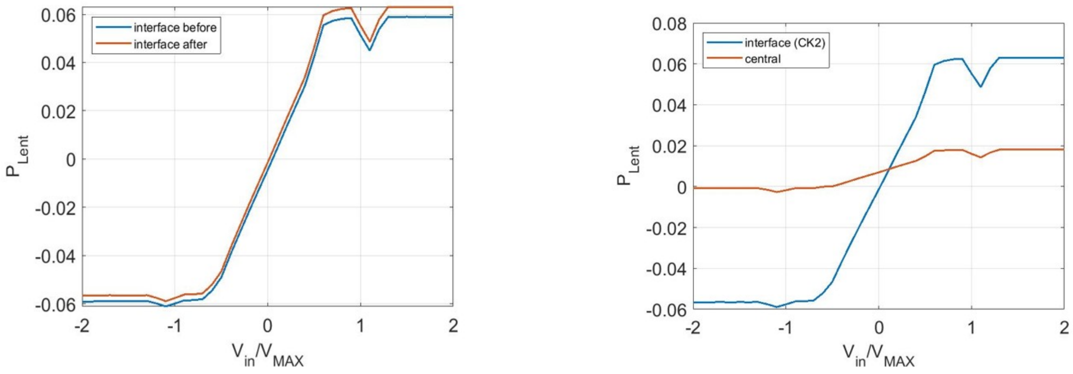
(b) $\alpha = 1.5V$

(c) $\alpha = 2V$

(d) $\alpha = 2V$

(e) $\alpha = 2.5V$

(f) $\alpha = 2.5V$



(g) $\alpha = 3V$

(h) $\alpha = 3V$

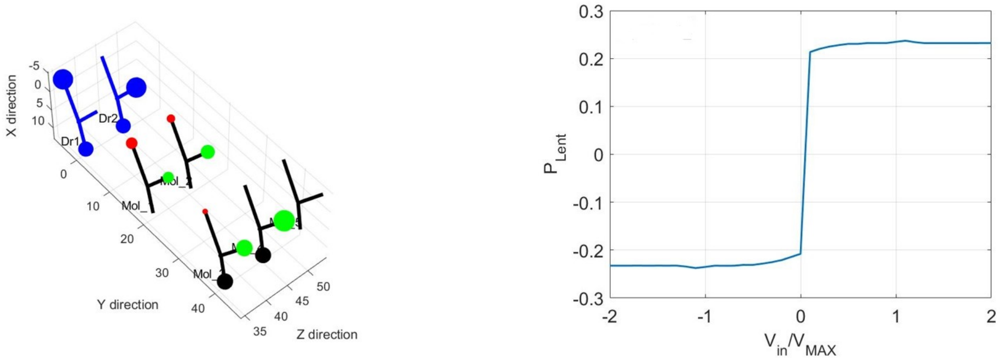
Fig. 4.16: Polarization curves for all the molecules in additional cell case. Interface m3 analysis

Two interesting primary considerations can be made about the above graphs. First, it is evident that for molecules characterized by higher values of saturation voltage, the effect of the back-propagation is almost null. The curves before and after the central cell activation superimpose almost perfectly in those cases, such as for $\alpha = 3V$.
This behavior is connected to the second important observation: for high values of α, the charge distribution in the central cell molecules becomes lower; in the last plots of figure 4.16 it can be either considered to be null. Interestingly, the polarization curves are not symmetric to the graph y-axis. In all the cases, the polarization shifts towards positive values.
This behavior is due to the asymmetric nature of the bis-ferrocene, which gives rise to asymmetric VACT and, therefore, to non-symmetric polarization curves. The bis-ferrocene central cell tends to distribute the charges in such a way to have positive polarization. This behavior is evident in the cases in which the saturation voltage of the interface molecules is high.

Vertical interfacing and crosstalk

The analysis of m1 and m2 interfaces requires a more detailed study due to some issues. First, the coupling is different from the one present along the y-direction; this is true for any FCN circuit. Moreover, it is different how the charges sense the electric field. It is important to remember that the electric field can be decomposed in contributions along the axis; of course, those can have various amplitudes. This concept is important because, as shown in the following, the crosstalk event is a real issue for molecular circuits built along the vertical direction.
Consider this situation: interface m1 characterized by $\alpha = 1.5V$ saturation voltage, and the involved layout is the *additional cell*.
The results are reported in figure 4.17.

Figure 4.17a shows a snapshot of the performed simulation, while 4.17b reports the polarization curve evaluated on the central cell.

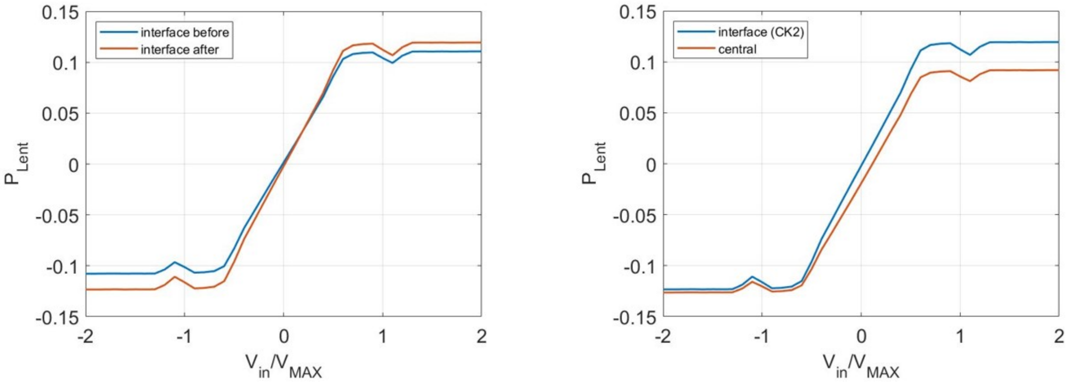


(a) (b)

Fig. 4.17: Interface m1. Additional cell layout with $\alpha = 1.5V$

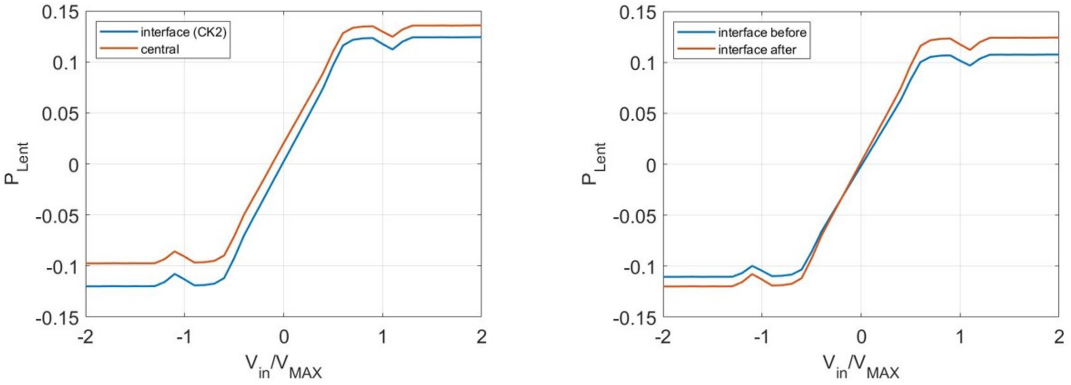
In particular, from this last one, it is possible to understand that, with the system as it is, the circuit does not work. The polarization reaches a positive saturation value for voltages higher than zero and viceversa for the negative side. Moreover, considering the neural implementation, such an interface cannot be characterized by a weight coefficient since there is no linear transition on the curve.
From a charge distribution point of view is useful to look at the snapshot reported on the left figure of 4.17. The charge is localized on the two lower dots along the z-axis due to the need to minimize Coulomb repulsion. The same problem can be noticed considering m2. In that situation, the charge will be localized on the two upper dots instead of the lower interface.
This is a big issue, particularly concerning the neural application under study.
This behavior can be thought of as molecular crosstalk, which is present in globally non-neutral molecules. The radial component of the electric field superimposes the propagation component. This radial component is not influencing the horizontal direction but only the vertical one. Consequently, the superposition of the previously mentioned contribution distributes the charges in the same direction, i.e., on corresponding dots of the two molecules forming the cell.
This analysis is important since it justifies the introduction of the counter-ion on the fourth dot of each molecule, thus using only globally neutral molecules on the circuit. The polarization curves obtained for the same interface with the presence of counter-ion are documented in figure 4.18.

The curves reported have similar characteristics to those presented in the analysis of interface m3. In this case, there is a clear change to the previous results: a linear region is present, and, therefore, the charge is better distributed among the dots of the central cell.
For what concerns the $P-V_{in}$ plots, an asymmetric behavior can be highlighted once more in the central cell polarization. Interestingly, this structure is not completely symmetric, opposite to the *drivers only* layout. Indeed, even though the crosstalk problems are solved using counter-ion, Coulomb repulsion has different effects if considered from the top or



(a) (b)

Fig. 4.18: Interface m1. Additional cell layout with $\alpha = 1.5V$, influence of the counter-ion

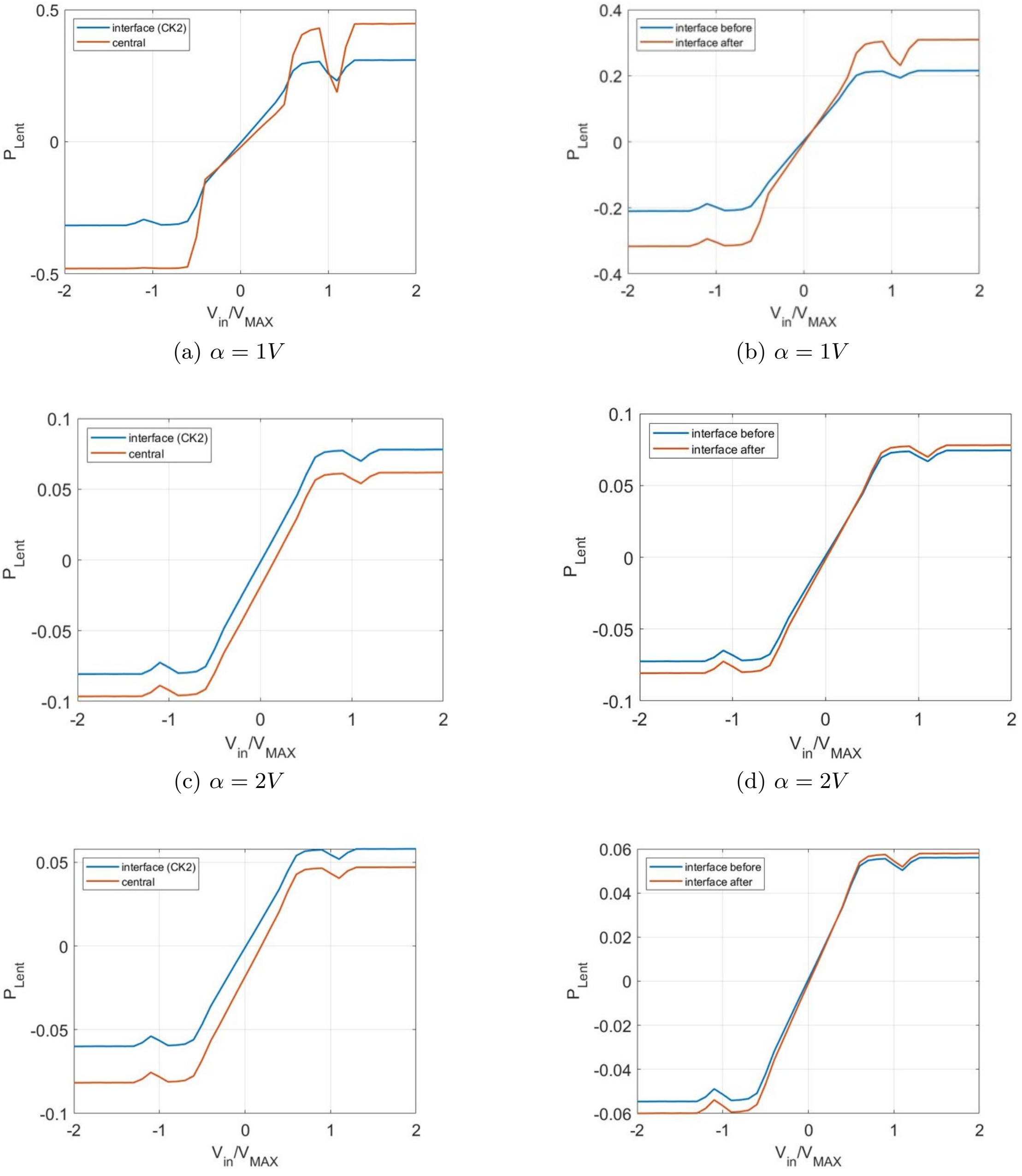


(a) (b)

Fig. 4.19: Interface m2. Additional cell layout with $\alpha = 1.5V$

the bottom along the z-axis. So, m1 interfaces enhance negative polarization values, while m2 does the opposite. This behavior implies that the weights are not equal for m2 and m1 interfaces.

In figure 4.20 are reported the polarization curves obtained in the analysis of interface m1 with the *additional cell* layout. As expected, the extreme values of the polarization become lower with the increase of the α parameter. In figure 4.19 is present the polarization curve for $\alpha = 1.5V$, obtained on the bottom interface.



(a) $\alpha = 1V$

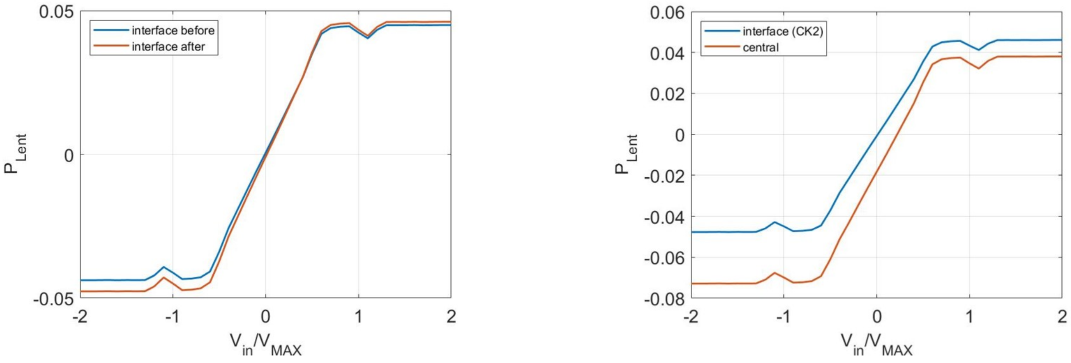
(b) $\alpha = 1V$

(c) $\alpha = 2V$

(d) $\alpha = 2V$

(e) $\alpha = 2.5V$

(f) $\alpha = 2.5V$



(g) $\alpha = 3V$

(h) $\alpha = 3V$

Fig. 4.20: Polarization curves for each molecule under analysis placed on interface m1

Charge distribution analysis

The previous section reports all the polarization curves obtained from the analysis of the upper interface, i.e., m1. It is interesting to notice how the one related to $\alpha = 1V$, reported in 4.20a, has a strange behavior if compared to the others. Moreover, this is the only case in which m1 and m2 results are not comparable since the m2 curve shows a smoother transition, as reported in figure 4.21. The issue that such behavior arises is that the weights are different and not comparable. For this reason, it was necessary to provide an in-depth analysis to understand the main reasons that caused this. Specifically, has been carried out a charge distribution analysis among the dots of the molecules in the central cell both for m1 and m2 cases. In figure 4.22 is reported a graphical representation of the dots positions from the interfaces point of view. In table 4.3 is presented the ideal combination of dots between which symmetry is expected if the same input voltage is assumed.

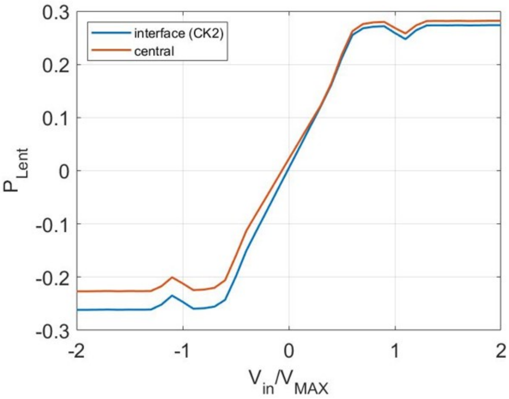


Fig. 4.21: Additional cell layout. Interface m2, $\alpha = 1V$

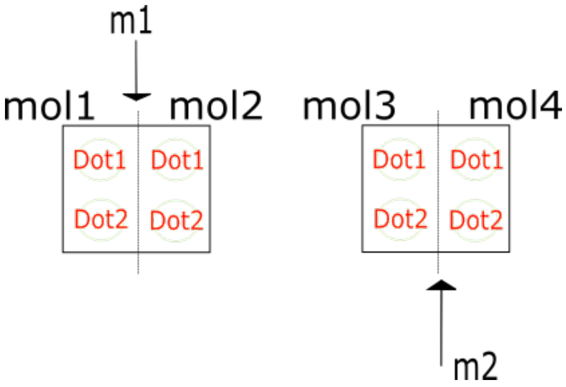


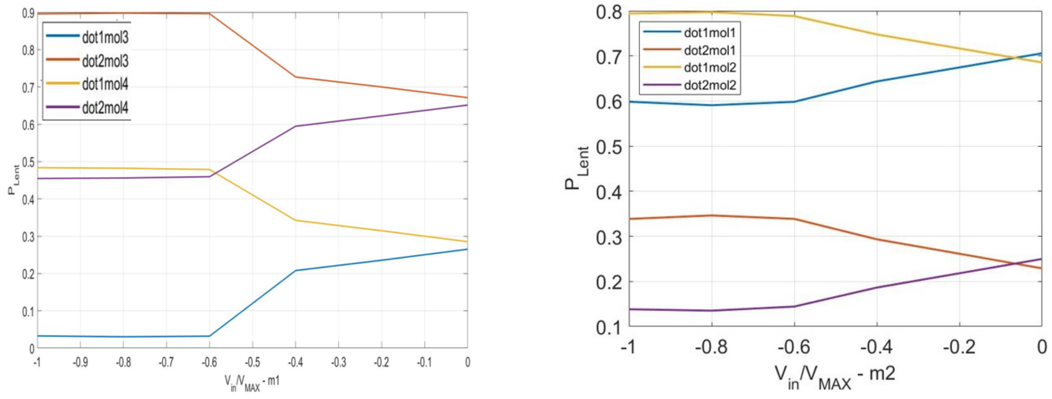
Fig. 4.22: Schematic representation: dots and molecules numbering

m1	m2
dot1mol1	dot2mol4
dot2mol1	dot1mol4
dot1mol2	dot2mol3
dot2mol2	dot1mol3

Table 4.3: Symmetries expected within dots of the central cell in m1 and m2 cases. With equal driving voltage, the charge distribution among those dots should be ideally equal

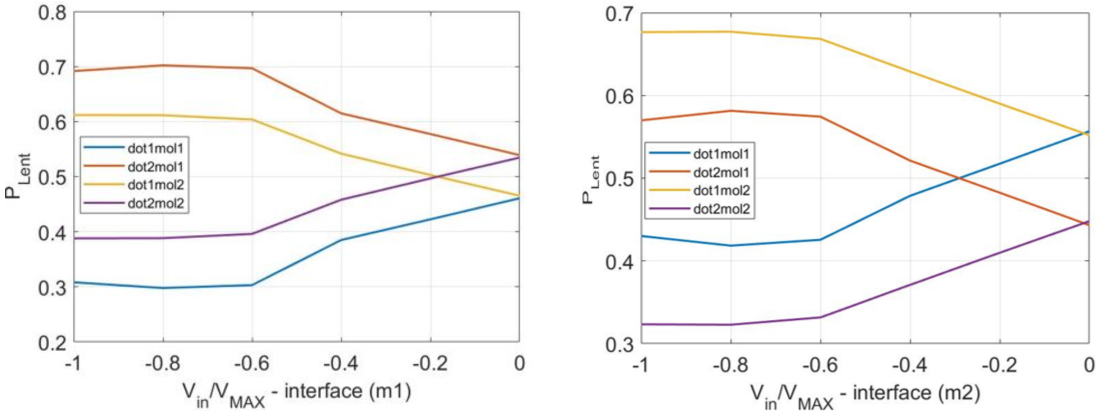
The charge distribution is evaluated on the interface and central cells for m2 and m1 cases. The input voltage range goes from -1V to 0V to focus the attention on the range in which the main problems occur. The results are reported in figures 4.23 and 4.24. The first presents the curve obtained from the analysis of the central cell, while the other shows the results obtained from the interface.
Looking at figure 4.23 it is visible a not proper charge separation on the molecule tagged as mol4 for input values ranging from -1V up to 0.6V. It is the same range in which the unwanted transition towards high polarization occurs in figure 4.20a. Looking at the counterpart of m2, reported on the right graph in 4.23, it is possible to appreciate a higher charge separation in the same input range. This situation occurs only in the central cell if molecules with saturation voltage equal to 1V are present at the interface. Indeed, the charge distribution is almost the same among corresponding dots in the case where $\alpha = 1.5V$ molecules are present at the interface of the bis-ferrocene central cell. This concept can also be understood from the comparable behavior of the corresponding polarization curves, reported in figures 4.18a and 4.19a.
Moreover, also for what concerns the interface charge distribution, no regions with null polarization can be detected either for m1 or m2. Therefore, the problem is just on the central bis-ferrocene cell with m1 connected to it. The reason behind this behavior can be a sum of causes. It is worth mentioning the possible presence of residual crosstalk effects,

the non-symmetric geometry and VACT of the bis-ferrocene molecule, and the absence of other molecules balancing the circuit, thus causing possible unwanted effects.
In conclusion, it is crucial to decide whether or not to consider the weights of this interface equal to the one derived for m2. Considering the voltage analysis and results in the following section, the more reasonable values have been found by considering the two interfaces as symmetric, thus having the same weight coefficients. This choice relegates the problems pointed out in this section mainly to the non-balanced structure of the circuit.



(a) m1 (b) m2

Fig. 4.23: Charge distribution plot for $V_{in} = [-1V, 0V]$. Central cell



(a) m1 (b) m2

Fig. 4.24: Charge distribution plot for $V_{in} = [-1V, 0V]$. Interface cell

Interface coefficients evaluation

As the last step, it is possible to evaluate the weight coefficients. The procedure is the same as in the previous cases. The polarization curves whose angular coefficient have been evaluated are the ones related to the central cell of the *additional cell* layout. In this way, the interface can be considered a building block that, once placed, has a different influence on the neuron body polarization, depending on the molecules involved.
The weights are reported in table 4.4, these are the final ones involved in all the following analyses and considered when applying the neuron's equation 3.2.

α	m1	m2	m3
1V	0.3543	0.3543	0.456
1.5V	0.1523	0.1631	0.0689
2V	0.1105	0.1169	0.0253
2.5V	0.0899	0.0945	0.0153
3V	0.0774	0.0811	0.0132

Table 4.4: Weight values: additional cell layout

Notice how, in this case, for α higher than 1.5V, interface m3 has a more negligible influence than m1 and m2. Moreover, the symmetric behavior of the vertical interfaces is valid only if the analysis carried out in the previous section is assumed. The corresponding coefficients are not equal but very close in amplitude for all the other molecules.

4.3.3 Voltage analysis for the *additional cell* case

The importance of the voltage analysis derives from the final objective that this work proposes. Evaluating the voltage behavior in different conditions of inputs and interfaces is essential since it provides a general idea about critical situations that can arise. In this section are provided some numerical analysis and comments about them. Before moving on, it is necessary to make a general and descriptive evaluation of the voltages and the influences involved in the structure. Until now, have been evaluated only separated interfaces; placing everything together implies that all the interfaces will have a specific impact on the polarization of the neighboring ones and viceversa. For this purpose, look at figure 4.25, in which are summarized the main voltage contributions and couplings. The light blue dots represent a possible inputs configuration; imagine at this point that the weights combine in such a way that the circuit works as a majority voter. Before the activation of the central cell, there is mutual influence along the diagonals; the black lines in the figure represent this. The results of these couplings are represented with the shadowed red-colored dots.

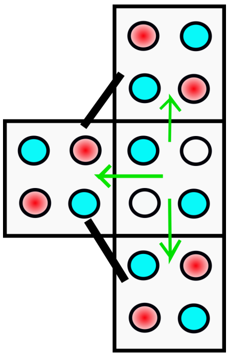


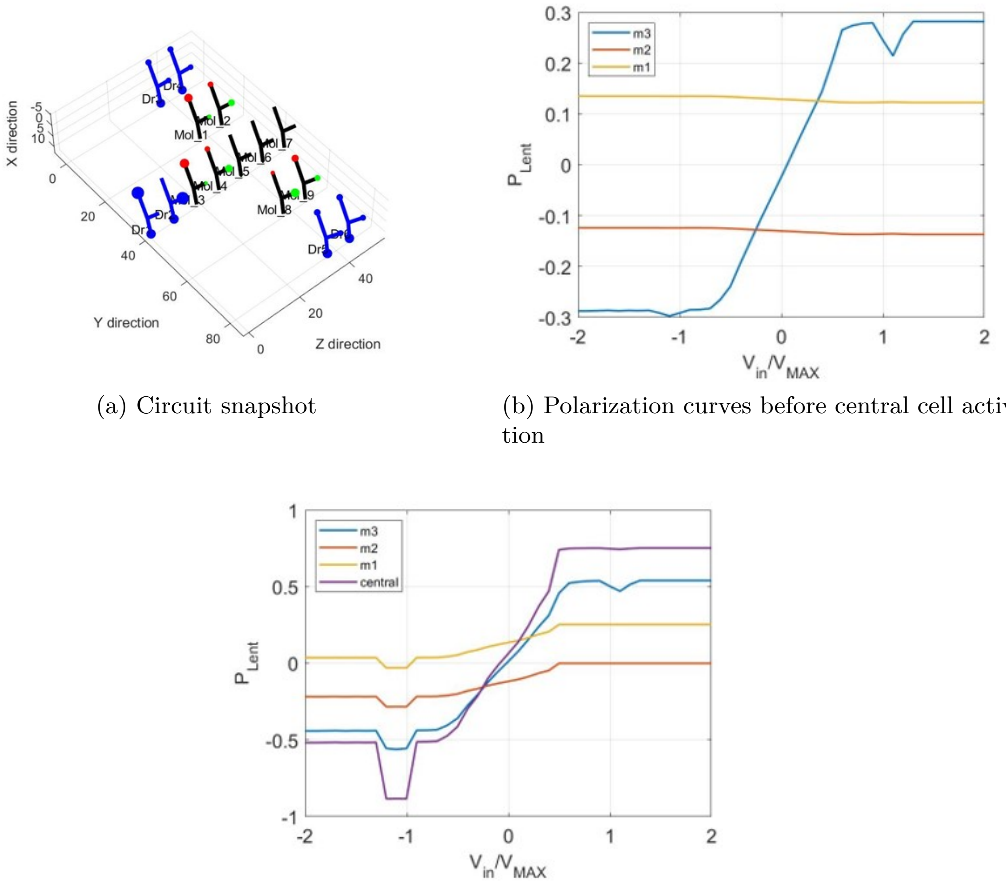
Fig. 4.25: Voltage influences among the circuit

Notice that the main problem in the particular situation under analysis is the influence within interfaces m3 and m2. Indeed, the cells lose polarization, and the charge tends to be distributed more homogeneously.
As soon as the central cell turns on, there will be back-propagation, and finally, the molecules reach the more stable and final configuration. Back-propagation is taken into account by the green arrows in figure 4.25 and it was already analyzed in the polarization curves presented in the previous section. Of course, the interfaces having same polarization as the one computed will experience an increase in their polarization, while the opposite happens in the reciprocal case.
Before moving on in the analysis it is essential to describe the choice regarding the signs of the input voltages at the interface. According to the weighted input formulation and the results, the best sign convention is the following one:

- Positive sign for input voltages providing a logic ‘0’
- Negative sign for input voltages providing a logic ‘1’

Figure 4.26 reports the first example of coupling within cells among the diagonals. These are highly indicative since only interface m3 is active, while the drivers on the other two are non-active. From 4.26a it is visible the influence that the activation of m3 has on the other two. All the interfaces present molecules with $\alpha = 1V$. These gain a certain amount of polarization due to the electric field lines coming from m3. Therefore, their polarization is different from zero, which should be expected considering only the influence of corresponding turned-off drivers.
These results are confirmed by the polarization curves reported in figure 4.26b.

From the proposed figures, it is possible to notice how the diagonal coupling influences the polarization on the non-active interfaces m1 and m2. It is interesting to evaluate the behavior of the induced polarization. From the plot in 4.26a, it is clear that m1 is subjected to a positive polarization, while on m2, a negative one is present. The important point is that for this circuit, the signs of the diagonally induced charge distributions are always the same on the whole input voltage range. In the case the vertical interfaces are active too, the resulting polarizations in the circuit have to consider both direct and diagonal couplings.



ù

(a) Circuit snapshot

(b) Polarization curves before central cell activation

(c) Polarization curves with central cell active

Fig. 4.26: Effects of the diagonal coupling and back-propagation

Moreover, in figure 4.26c, can be appreciated the back-propagation effects deriving from the configuration of the charges in the central cell. In this case, all three interfaces follow the exact behavior of the central cell.

The characterization of the diagonal coupling should be done for all the interfaces, and with all the molecules involved, such work would be necessary for the final implementation and in-depth understanding of all the possible scenarios. However, this is out of the scope of the present work; at this point, it is enough to understand the possible impact of this effect on neural computation.

In the last part of this section, two other examples present different possible situations. In the first one, the configuration of the circuit is the following:

- **Interface m3**: input voltage sweep from -2V to +2V, $\alpha = 1.5V$
- **Interface m2**: $V_{in_2} = 0.7V$, $\alpha = 2V$

- **Interface m1**: $V_{in_1} = -0.8V$, $\alpha = 2V$

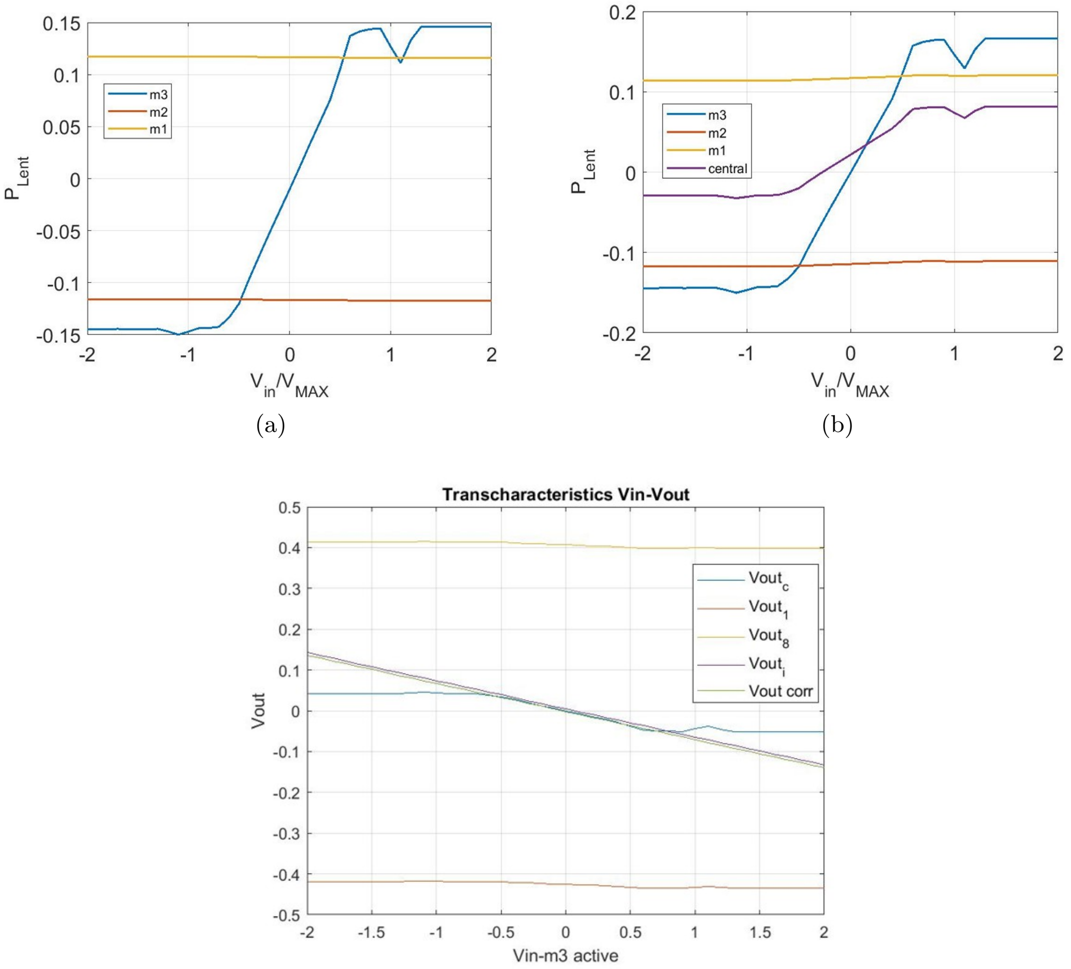
The results are reported in figures 4.27, which shows the polarization curve at the interfaces before and after the central cell activation. The third one reports a voltage analysis on the whole input range. The curves must be read in this way: V_{out_1} is the voltage on the first molecule of m1; V_{out_8} is, instead, referring to the interface m3; finally V_{out_c} is the voltage evaluated on the left molecule of the central cell, as usual.
Related to this, some interesting considerations can be provided. First, it is possible to appreciate how, in the linear region of the V_{out} curve, the errors within the expected ideal values given by 3.2 and the obtained ones are minimal; the difference value within expected and obtained voltages is going to referred as prediction error. This result is due to the particular combination of interfaces and inputs chosen for this simulation. Indeed, the provided polarization is low for all the interfaces, meaning it is possible to neglect the spurious contributions from other interfaces through diagonal or vertical coupling. Moreover, the fact that the curves are close indicates the quality of the weights evaluated in the previous section. Thanks to this simulation, it is possible to state that it will be possible to reasonably predict the voltage on the first molecule of the central cell and, consequently, its final configuration. Similar reasoning can be applied to the polarization curves reported in figures 4.27a and 4.27b. About the first one, some comparisons can be made with the results presented in 4.26b. The polarization values obtained on the vertical interfaces are comparable. The main difference is that similar results were obtained only with the influence of m3 along the diagonals in the previous case. In this second case, the upper and lower drivers are active, so the polarization achieved through their influence must be summed to the one obtained through diagonal coupling. This last contribution is smaller than in the previous case. That is mainly due to the presence of molecules characterized by α values higher than one. These, as already analyzed, are more difficult to be polarized, and the charge distribution among their dots is lower, thus obtaining a cell less susceptible to external influences. A third and last example is reported in figures 4.28. The new configuration is:

- **Interface m3**: input voltage sweep from -2V to +2V, $\alpha = 1.5V$
- **Interface m2**: $V_{in_2} = 0.7V$, $\alpha = 2V$
- **Interface m1**: $V_{in_1} = -0.7V$, $\alpha = 1.5V$

Notice that the green line in figure 4.28c represents the ideal line shifted down by an amount equal to the mean error present in the linear region of the V_{out} curve. This case is interesting given the presence of different molecules on the two cells along the z-direction of the circuit. As can be noticed from the curves, the polarization is not symmetric as in the last analyzed simulation but more significant on the cell having a lower α value, i.e., m1. The voltage analysis still provides good results, even though the error is greater than the one measured before. Moreover, it is possible to demonstrate that, with proper weight selection, the least present logical configuration can be transported to the neuron output, reversing the usual behavior of the majority voter circuit. An example of this is the following:

- **Interface m3**: $\alpha = 1.5V, V_{in_3} = 0.4V$
- **Interface m2**: $\alpha = 1.5V, V_{in_2} = 0.5V$
- **Interface m1**: $\alpha = 1V, V_{in_1} = -0.7V$

As anticipated, m1 is the prevailing interface in this case, and, even if its configuration is less present than the first two, this is the one that propagates to the output. The voltage obtained from the simulations on the first molecule of the central cell is equal to -0.1753V, with a prediction error equal to 0.0364 in absolute value. Notice that this result, particularly the low prediction error, justifies the choice related to the weight for the $\alpha = 1V$ molecules on interface m1.

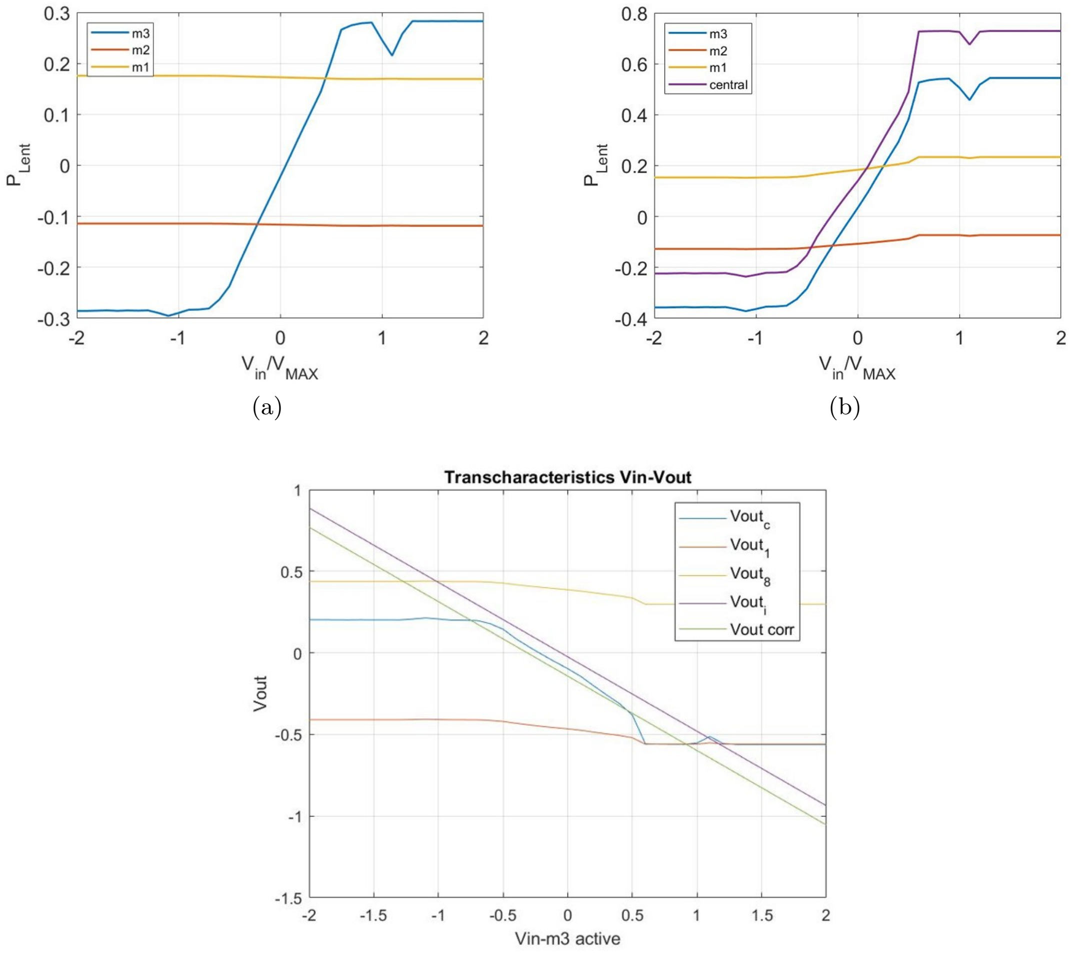


(a)

(b)

(c)

Fig. 4.27: Voltage and polarization results for the additional cell complete structure. Second example.



(a)

(b)

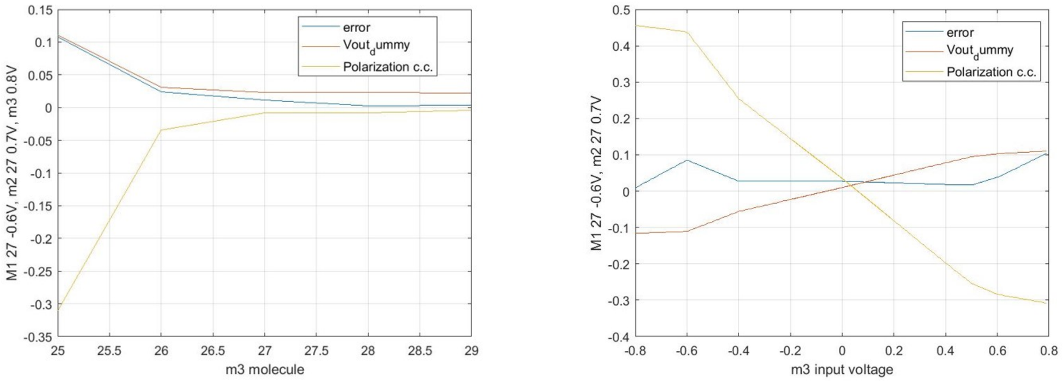
(c)

Fig. 4.28: Voltage and polarization results for the additional cell complete structure. Third example.

4.3.4 Parametric analysis

Many simulations were performed to define some basic general rules of workability of the circuit. Summing up the results, it is possible to provide parametric analyses, in which the parameter can be either the input voltage to a certain interface or the molecules present. The two graphs in 4.29 report examples for each type of study. Specifically, in the plot shown in figure 4.29a, the parameter is the α value of the molecules placed on interface m3. The input voltage V_{in_3} is set to 0.8V. The other two have fixed-configuration: m1 is characterized by $\alpha = 2V$ and V_{in_1} = -0.6V; m2 instead shows $\alpha = 2V$ and V_{in_2} = 0.7V.
The values reported on the graph are the errors obtained concerning the ideal calculation, the polarization of the central cell, and the voltage obtained on the dummy molecule. For

what concerns the first two, it is possible to verify what was already understood from simpler layouts: the lower, in absolute terms, the polarization of the central cell, the lower the displacement from ideality. Indeed, the error curve increases with increasing α parameter, and the polarization becomes lower in absolute value.
The situation reported in 4.29b shows some different behavior. In this case, the parameter is the input voltage on the interface m3, which is set with $\alpha = 1V$ molecules. The other interfaces have the following setup: $\alpha = 2\text{V}$, $V_{in_2} = 0.7V$ on m2; $\alpha = 2V, V_{in_1} = -0.6V$ on m1. This second example is of particular interest since it shows an almost flat behavior of the error value around 0 over the whole input voltage range analyzed, which goes from -0.8V to 0.8V. The polarization, instead, has a straight behavior from positive to negative values. The conclusion deduced from this second set of results is that this configuration is extremely stable for any voltage value at the input of m3. Such a behavior derives from the presence of one strong interface while the other two contribute not significantly to the central cell final polarization and, moreover, have opposite voltages and same α. This way, the computational cell mainly follows the behavior of m3.



(a)

(b)

Fig. 4.29: Parametric analysis of additional cell layout

Several analyses of the same type can be provided to define trend behavior for different configurations of the interface molecules or the input voltage coming from a specific direction. In general, it is possible to highlight similar considerations to those found before: the behavior of the error is not easy to predict and depends on the mutual influences present on the structure and the active molecules. In general, with this type of system is difficult to work out of linearity due to the low polarization of the interfaces. However, even if small, the evaluated error can lead to wrong predictions of the final logic behavior of the neuron. These considerations are going to be analyzed in detail in the following chapters. At this point, a general idea of the possible scenarios is enough to state that such a neuron can work if inserted into a network.

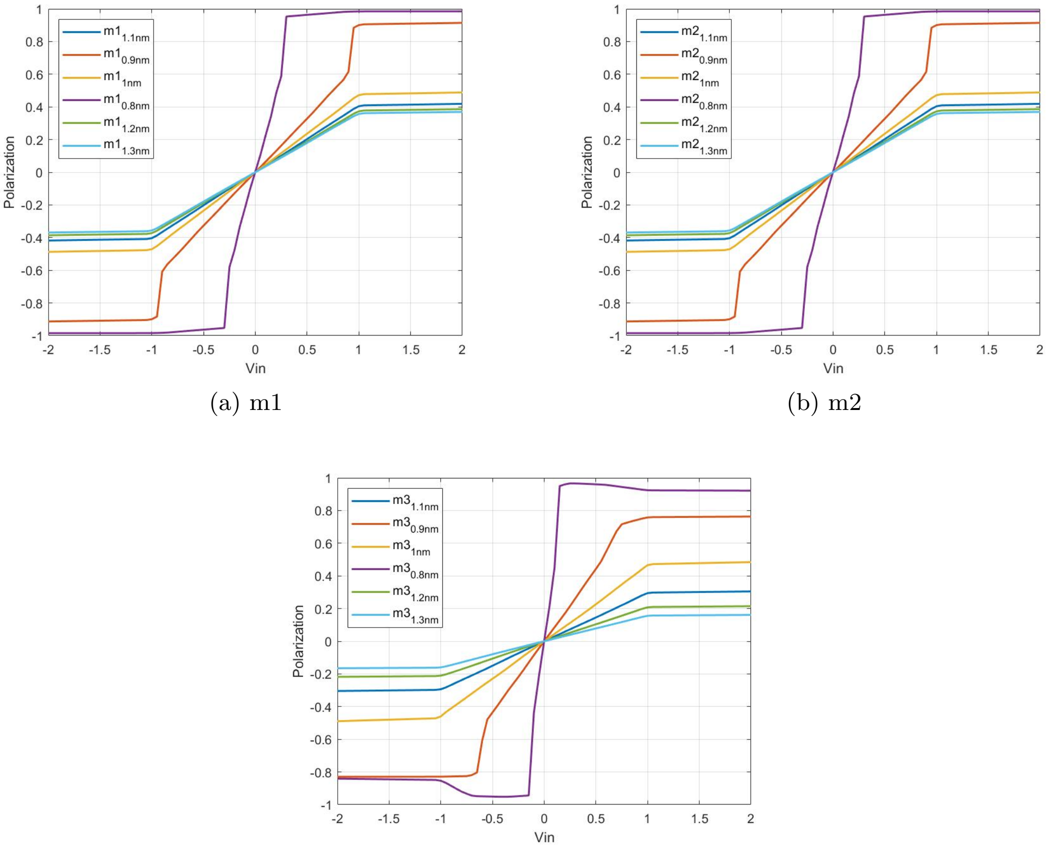
Influence of intermolecular distance

At the end of the first chapter were given some hints related to the technological implementation of FCN. As in any fabrication process in electronics, the different steps to which the device is subjected have some tolerances and limited resolution. For this reason, process variations can affect the final device. One parameter with a significant influence is the distance within the molecules, i.e., the intermolecular distance. When dealing with a new type of circuit, it is crucial to characterize different values of the most critical geometrical parameters. Up to now, the intermolecular distance along the longitudinal direction was always set equal to 0.9nm. This section presents parametric analyses to characterize the system for different spaces within the molecules.
Besides the importance of a formal numerical evaluation, the general trend has to be analyzed; for this reason, to stay in an easy situation, the chosen layout is the *drivers only*, and the parametrized distance is the one along the y-direction.
The three sets of plots in figures 4.30a, 4.30b and 4.30c reports all the polarization curves obtained studying m3, m2 and m1 for different values of separation within the molecules. These are characterized, in this first case, by $\alpha = 1.5V$. The range for the distance change is [0.8nm, 1.3nm].

In all the graphs reported, it is possible to notice how the increase of the distance parameter decreases the weight of the molecules. On the other end, its reduction tends to drive the bis-ferrocene into saturation. These behaviors are evident in the curves related to d = 0.9nm. For separation values along the y-direction higher than 1nm, we can highlight a sort of saturation in the behavior of the different curves. Interestingly, in m3, the curves reach this value slower than in the other two cases. This is reasonable since the distance variation applied for these simulations involves to a lesser extent the cells on the vertical interfaces. Indeed, each molecule is getting closer or farther to the other one in the same cell, but the distance from the neighboring cell is unchanged since it lies in the z-direction. Having in mind the steps performed on d = 0.9nm, it is possible to follow the same path to evaluate the trend of the parameters defining the polarization curves for each interface. The results are reported in figures 4.31a, 4.31b, notice that just the curves for m2 and m3 are reported since the *driver only* layout maintain the symmetry within m1 and m2. Each graph reports three curves:

- m: this is the weight coefficient of the interface
- V_{sat}: this is the voltage value at which the polarization of the central cell reaches saturation
- P_{max}: this is the maximum value of the polarization on the central cell.

The more interesting profile is the one related to the weight coefficients. As could have been understood from the polarization analysis, the curve decreases with distance. The weights for separation values lower than 0.85nm are higher than one. It is a clear indication that for any value present at the cell input, with such a small distance, the central cell will saturate. For spaces larger than 1nm, the coupling within the molecules tends to become null; consequently, the curve tends to zero.

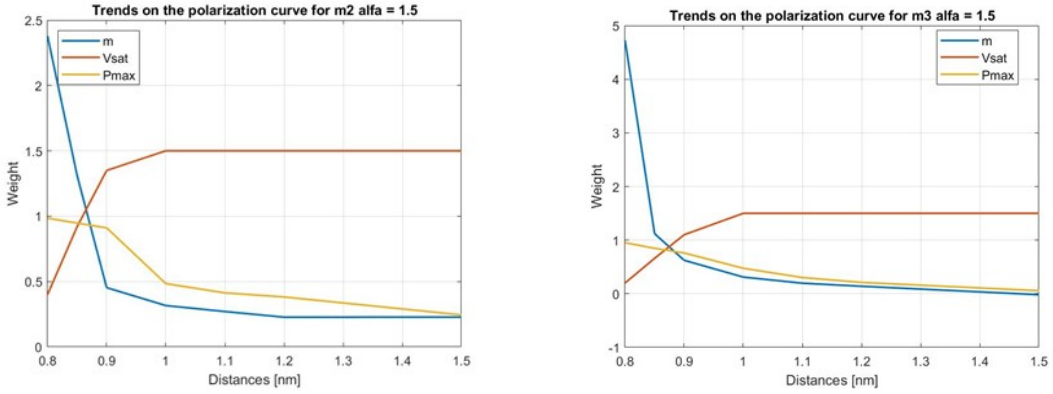


(a) m1

(b) m2

(c) m3

Fig. 4.30: Polarization curves for different values of intermolecular distance. $\alpha = 1.5V$

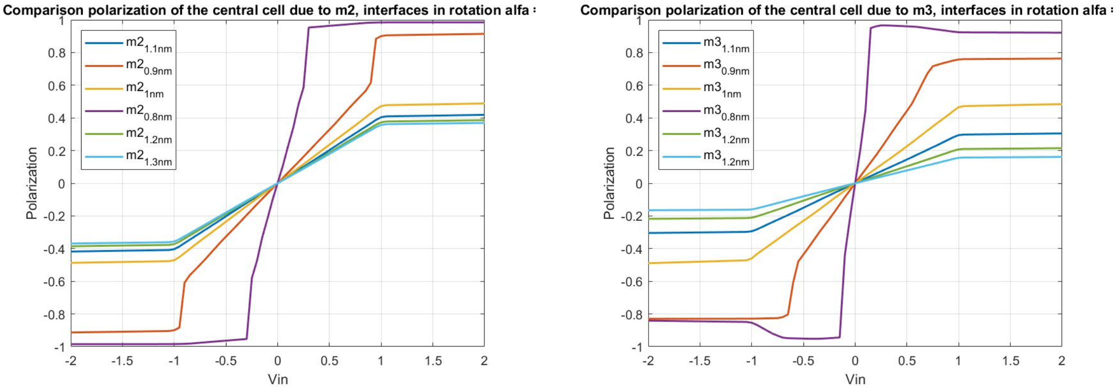


(a) m1

(b) m2

Fig. 4.31: Parameter curve vs distance variation. $\alpha = 1.5V$

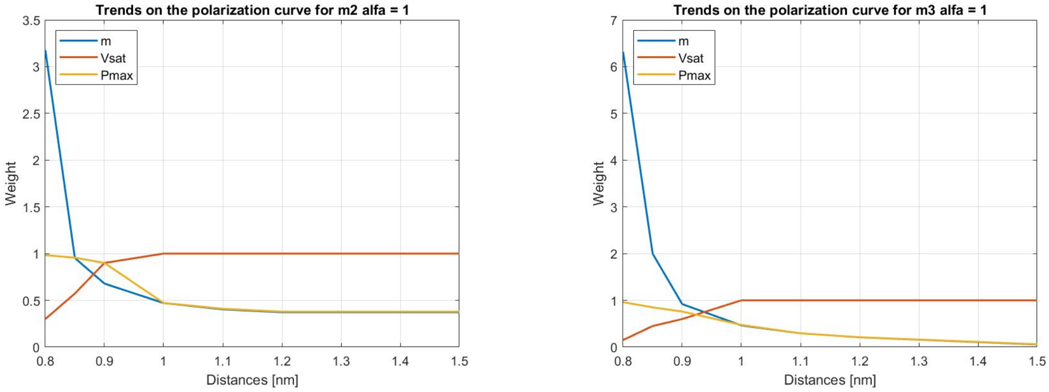
From a mathematical point of view, this trend can be evaluated as a decaying exponential, and this is reasonable since, from the standard rules of electrostatics, is well-known that the influence of an electric field on a certain point in space is inversely proportional to the square root of the distance. In the same way have been characterized the interfaces for other molecules. For instance, the results for $\alpha = 1V$ are reported in the sets of figures 4.32 and 4.33. In these can be noticed the same trends as in the previous graphs. Therefore, it is correct to state that, whatever the molecule under study, the results of the distance parametric analysis will provide the same result.



(a) m2

(b) m3

Fig. 4.32: Polarization curves vs distance variation. $\alpha = 1V$



(a) m2

(b) m3

Fig. 4.33: Curve fitting $\alpha = 1V$

This analysis can be considered enough. Obviously can be enlarged to understand better the *additional cell* layout behavior and, therefore, of the complete neuron. However, the essential point of this work was to set a limit to the intermolecular distance that should be adopted. This can be defined in the range between 0.9nm and 1nm.

Chapter 5

Propagation analysis for molecular FCN neural networks

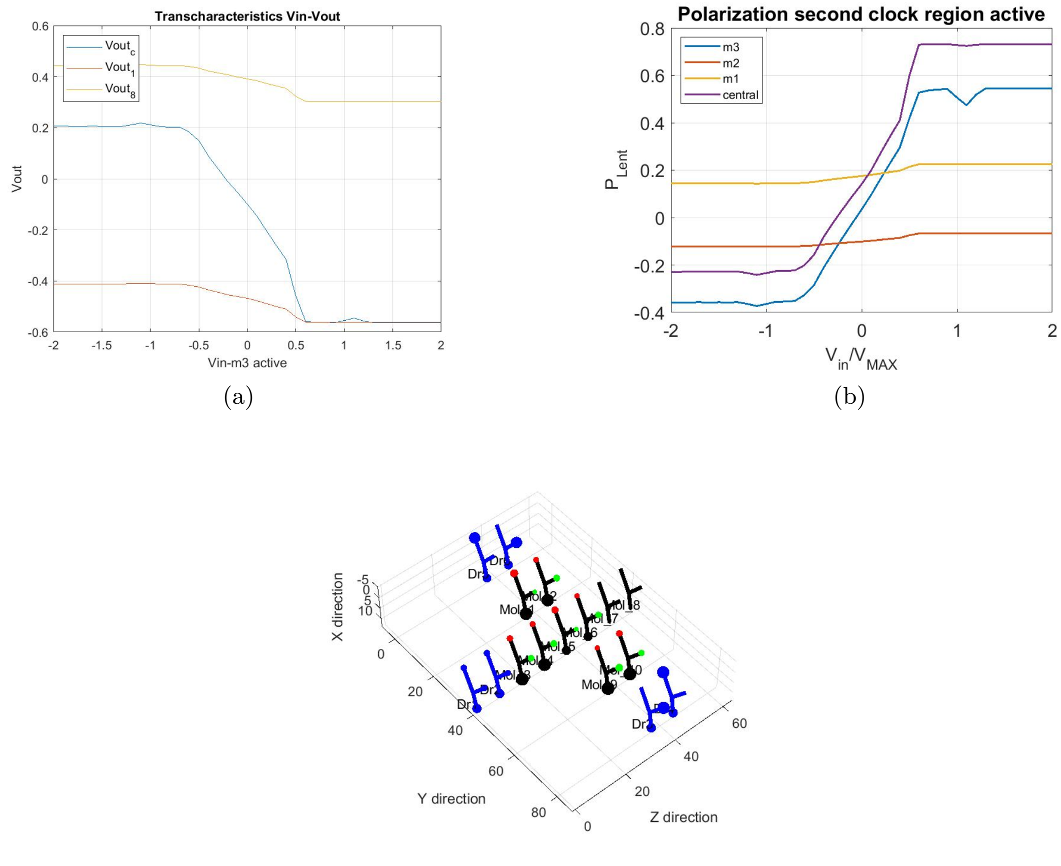
The information transport is a critical point in developing any electronic technology. The main requirements are that information is not degraded along the wire on which is transmitted, and the logic value at the end of the wire is congruent with the one sent. In order to propose a solution for the realization of neural networks in molecular FCN, it is mandatory to study how the molecular wire propagates the logic values.
Molecular wires are analyzed in depth in literature [22]. However, the main focus of this chapter is the characterization of the output wire connected to the neuron in terms of molecules and clock region distribution. In this way, the final result will be inserted in the neural network, connected to the already studied neuron cell.

5.1 Output wire characterization

The first parameter under test is the distance within the molecules. The starting layout is the one defined as *additional cell*, with d = 0.9nm along the y axis. A first example is reported in figure 5.1. These results are obtained through a simulation involving a sweep on m3 = 1V, while $V_{in_2} = 0.7V$ on m2 = $2V$ and $V_{in_1} = -0.7V$ on m1 = 1.5V. The clock regions are organized as follows:

- clock region one hosts the interface cells
- clock region two his occupied by the central cell of the neuron
- clock region three influences the output cell only

The clock signals have been configured according to rules of the adiabatic propagation [14].



(a)

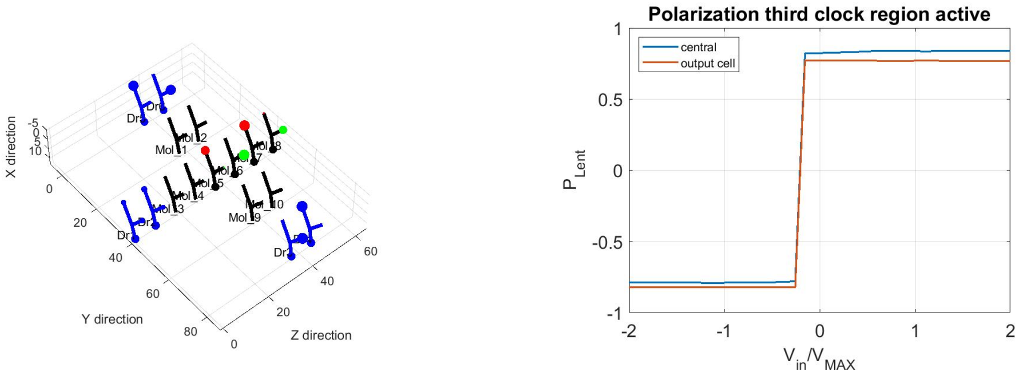
(b)

(c)

Fig. 5.1: Addition of the output cell. m3 = 1V, m2 = 2V V_{in_2} = 0.7V, m1 = 1.5V V_{in_1} = 0.7V

Figure 5.1 shows the voltage and charge distribution results as soon as the central cell is active. It is possible to notice a strongly non-symmetric behavior of the V_{out} values in the input voltage range. At this point, the same considerations carried out in chapter 4 can be considered valid. The situation changes as soon as the output cell of bis-ferrocene is activated by the influence of the third clock signal; the results are reported in figure 5.2. The main evident behavior is that the system polarization saturates for any input voltage value. The polarization, indeed, assumes values equal either to -1 or +1, with a vertical transition between the two. Notice that the voltage at which the transition occurs is not at 0V; this is due to the non-symmetric behavior of bis-ferrocene VACT. It is possible to conclude this first analysis with 0.9nm distance, saying that it is impossible to propagate the information linearly. The final system will be fully digital, working only with two binary values.

In hardware implementation of ANN, mainly the analog ones, the signal levels going through the network can have any voltage value. An analog-like performance of the



(a)

(b)

Fig. 5.2: Addition of the output cell: activation of the output cell

molecular circuit would increase the possibilities related to this solution. On the other hand, if a digital system is the goal, it is enough to work with analog values on the driver side and then saturate the charge distribution at the beginning of the output wire to transport logical '0' and '1'.

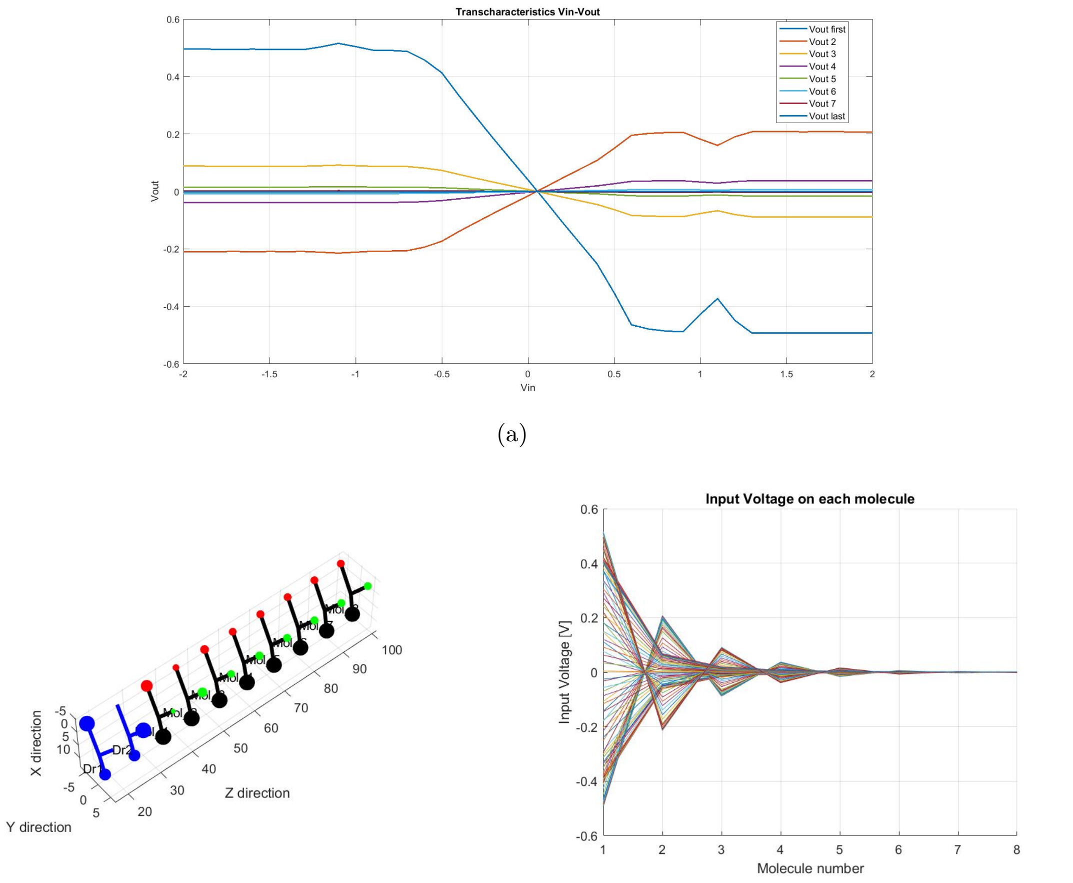
To develop an analog system with molecules, it is needed to work in the linear region of the characteristics, preventing the charge from complete distribution. For this reason, such a result is not easy to obtain. In the following sections of this chapter, different parameters and configurations are analyzed to understand the possibility of implementing an analog neural network with the molecular FCN paradigm.

5.1.1 Saturation voltage variation

Following the results reported so far, something must be changed to have non-saturated propagation, either in the structure geometry or in the molecule selection. Indeed, there are mainly two parameters that it is possible to change to try to develop an analog propagation: the saturation voltage of the molecules and the intermolecular distance. Concerning the first, it is possible to consider the lowest value of the ones introduced by this work, i.e., $\alpha = 1V$.

At this point, only the output wire is simulated since the main focus is on how the information propagates, thus decreasing the computational weight of each simulation. Therefore, the driver represents the computed information at the input of the wire, which generally could have values between -0.8V and 0.8V.

The results of the simulation are reported in figure 5.3. Figure 5.3a is particularly interesting since it reports the V_{out} curves on each molecule of the wire. Through this figure of merit, it is possible to evaluate the decaying behavior of the voltages along the molecular wire. In this case, with molecules having a saturation voltage double to that of the bisferrocene, the propagation cannot occur, and the information is inevitably lost after the first cell. This behavior is confirmed by the two other figures of merit in 5.3, in particular the one reporting the charge distribution along the wire.



(a)

(b)

(c)

Fig. 5.3: Molecular wire made of $\alpha = 1$V molecules

The above considerations allow to state that the change of the saturation voltage, i.e., of the molecules, adopted in the realization of the molecular output wire is not a solution at this point. The main reason for that is the extremely fast decay of the information during the propagation, thus not allowing for a correct read-out.

5.1.2 Change of the intermolecular distance

At the beginning of this chapter, it was shown that there is no possibility of working in linear conditions for separation values lower than 1nm. Being discarded the possibility of changing the molecules, the remaining parameter is the intermolecular distance, whose influence has already been explained at the end of chapter four.
The structure that is going to be analyzed is the following one:

- the driver simulates the presence of an input to the central cell, thus representing the combination of the input voltages from the interfaces

- the first cell of the molecular wire works as the central cell of the neuron and is located in the first clock region
- the other molecules of the output represent the output wire and are controlled by the second clock zone
- given the results obtained in the previous section, all the molecules involved are bis-ferrocenes

What will be analyzed is the type of propagation according to different distances *d* and different lengths of the wire, i.e., different number of molecules. For each case evaluated with d = 1nm, the maximum voltages that can be applied at the input of the wire before it reaches the saturation condition are considered. It is also interesting to compare the results with the d = 0.9nm situation. Some preliminary analyses are reported in figure 5.4.

The curves can be defined as follows:

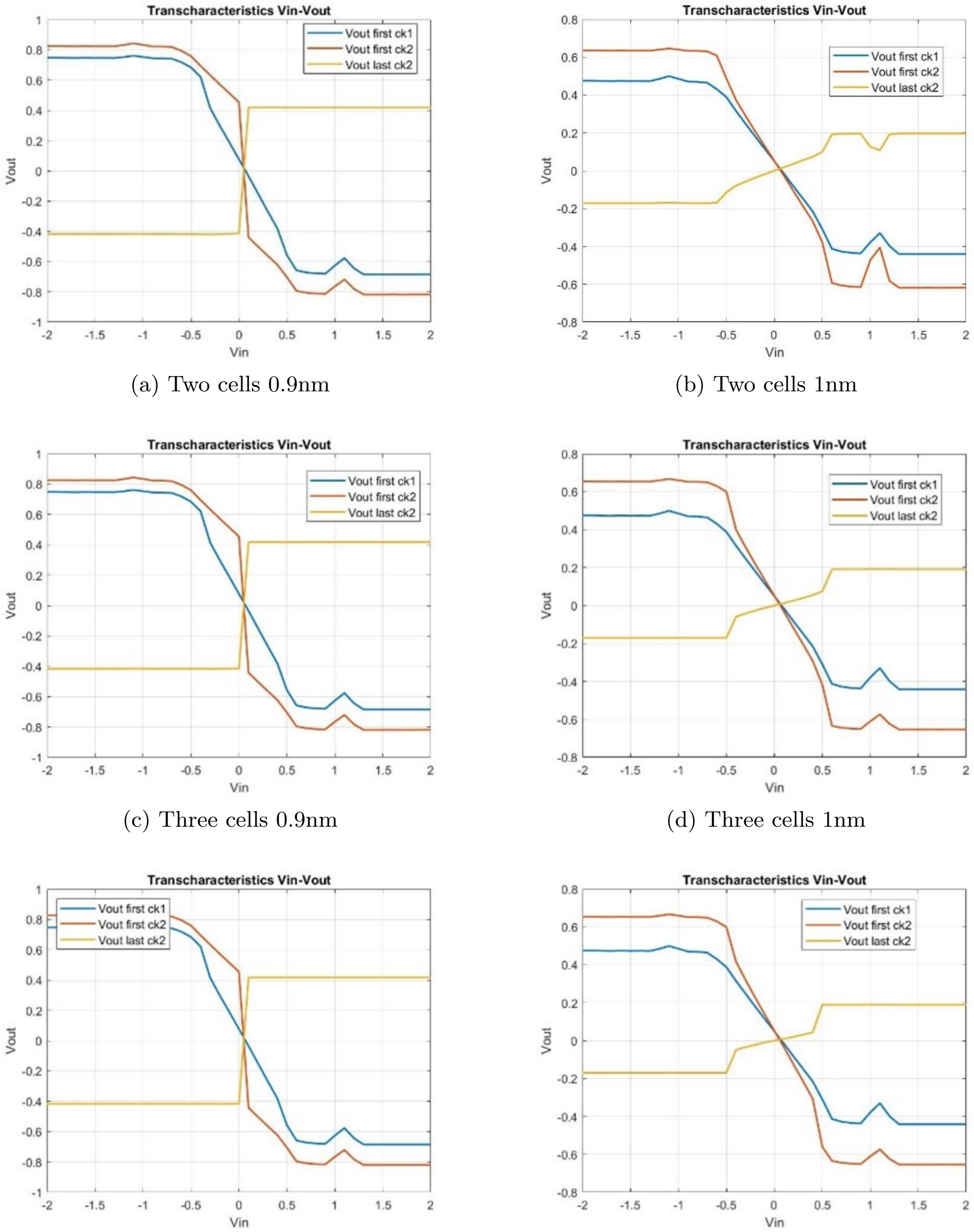
- V_{out} **first ck1**: it is the output voltage on the first cell of the molecular wire when the second region is not active yet, so the output voltage at the output of the neuron body
- V_{out} **first ck2**: is the output voltage on the first molecule of the output wire when the second clock region is active. It is possible to highlight the effects of back-propagation by means of this curve
- V_{out} **last ck2**: is the voltage evaluated on the last molecule of the output wire when the second clock region is active. It is interesting since it will represent the input to the next neuron interface in the complete structure. A too low value will translate into a very small interface polarization.

Looking at the results related to d = 0.9nm, it is possible to notice that these are very similar, confirming the results already found: it is impossible to have linear propagation with such separation. For what concerns the results for *d* equal to 1nm a linear transition is present; it suggests the possibility of working in a non-saturated region and transporting analog information. Notice that another condition to have linear propagation is that the voltage computed by the neuron and given in input to the wire is small enough to prevent the wire from saturating.

Given the results, it is possible to say that the situation reported so far can be associated with very local interconnections from a technological point of view. Local connections in FCN technologies have to be handled with care due to the possible presence of spurious influences deriving from electric field lines propagating through the circuit.

5.1.3 Wire length parametric analysis

Consider a four cells wire. Notice that the first two molecules represent the central cell of the neuron and, therefore, are activated by the first clock signal. Consequently, just six molecules form the wire. Before moving on in the analysis of the results, it is helpful



(a) Two cells 0.9nm

(b) Two cells 1nm

(c) Three cells 0.9nm

(d) Three cells 1nm

(e) Four cells 0.9nm

(f) Four cells 1nm

Fig. 5.4: Polarization and output voltage characteristics of molecular wires with two clock regions, variable distance and variable number of molecules

to introduce two parameters, namely β_+ and β_-. These are, respectively, the maximum and the minimum values that can be applied at the input of the wire before it reaches the saturation condition.
In figure 5.5 are reported the curves representing the output voltage for each molecule in the chain for the four cells wire.

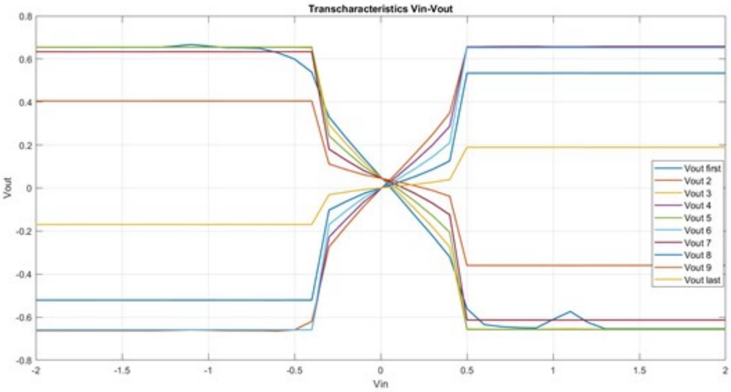


Fig. 5.5: Output voltages on all the molecules of a molecular wire with d=1nm

From the figure above, it is evident the presence of a linear region. The maximum input values are $\beta_+ = 0.4V$ and $\beta_- = -0.4V$. Due to the linear propagation and border effects, the voltage at the output of each molecule decreases with the increase of the distance from the input driver. This behavior is correct since, without the complete polarization of the charge distribution, the voltage gain of each cell is lower than one, meaning that the voltage decreases from one cell to the following. To provide some more parametric study, figure 5.6 presents the same results but with increased intermolecular distance, equal to 1.1nm. It is evident that there is no propagation in this case, and the results are similar to the case in which $\alpha = 1V$ molecules were used to build the output chain. The first conclusion is that the only possible value of molecule separation is 1nm to work in the analog domain. It also confirms the analysis related to the change of the weights, highlighting that for too high separation, the weights of the interfaces would become too low to make the system work correctly.

Restoring back the distance to 1nm, it is possible to provide other more profound studies considering applying single inputs to the wire. In figure 5.7 are reported the results related to the situation in which the input voltage is set at -0.4V; as can be expected, the wire works in the linear domain. In figure 5.7a are reported the output voltages at each molecule; it is evident the lossy behavior present in this type of solution.

Also related to this simulation, it is interesting to evaluate the polarization on the last cell of the wire, which is equal to 0.0662. From this value, it is possible to state that the charge separation on the last cell is practically absent. Therefore, the arising problem is the low influence on the interface cell that will be connected at the end of the wire. Border effects and linear propagation combined are such that the information is still correct from an overall charge distribution point of view, but the exact voltage value is completely

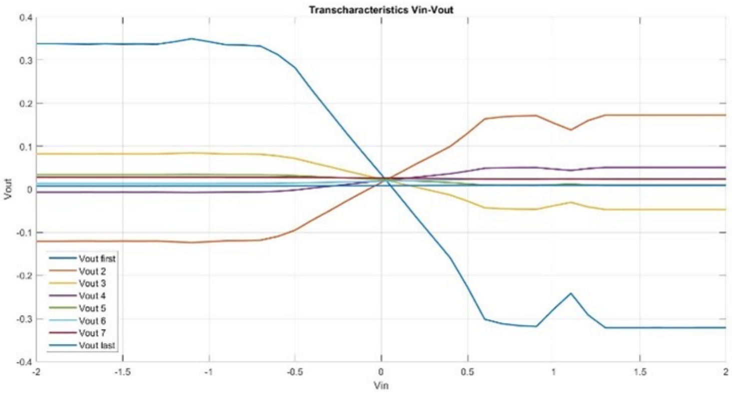
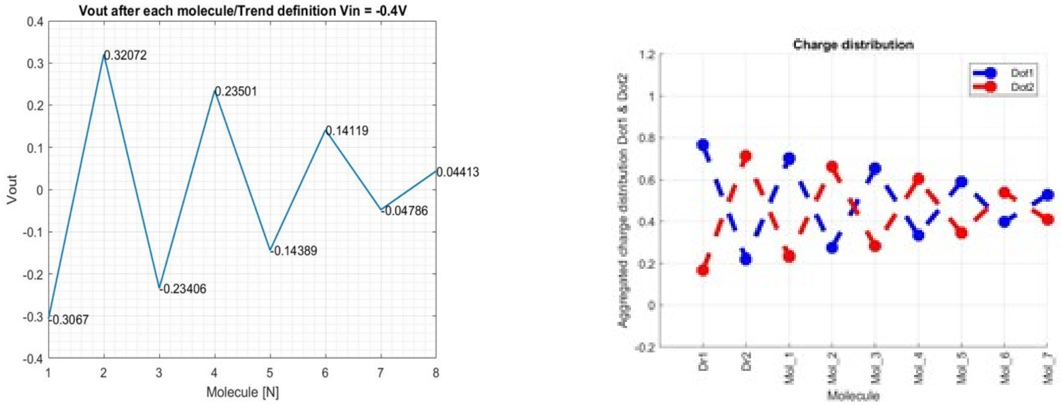


Fig. 5.6: Output voltages on all the molecules of a molecular wire with d=1.1nm



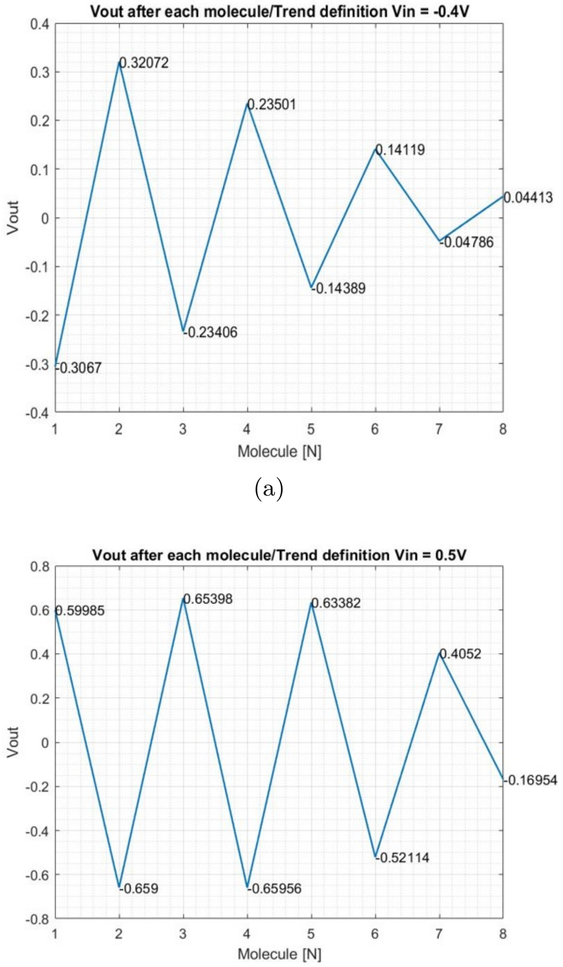
(a)

(b)

Fig. 5.7: Output voltages and charge distribution on an eight molecules wire subjected to an input voltage $Vin = -0.4V$

lost. It is a major problem that has to be solved if the goal is to achieve a working analog propagation. Notice that the lossy behavior and border effects are independent of the length of the wire. Whatever N, provided that it does not generate saturation, the voltage will be small on the last molecule of the wire, and the system will lose the initial analog information.

In figure 5.8a is reported the case for $V_{in} = 0.4V$, notice that the problems just pointed out are present also in this case. However, the voltage values are higher, and the polarization level on the last cell is equal to -0.1439. As a final result, figure 5.8b reports the voltage trend curve for $V_{in} = 0.5V$, for such an input voltage the molecules in the wire saturate. The consequence is that the molecules have equal in magnitude and opposite in sign output voltages, around 0.6V. Once again, the initial voltage value is lost, but the polarization on the last cell has a useful value equal to -0.4176.



(a)

(b)

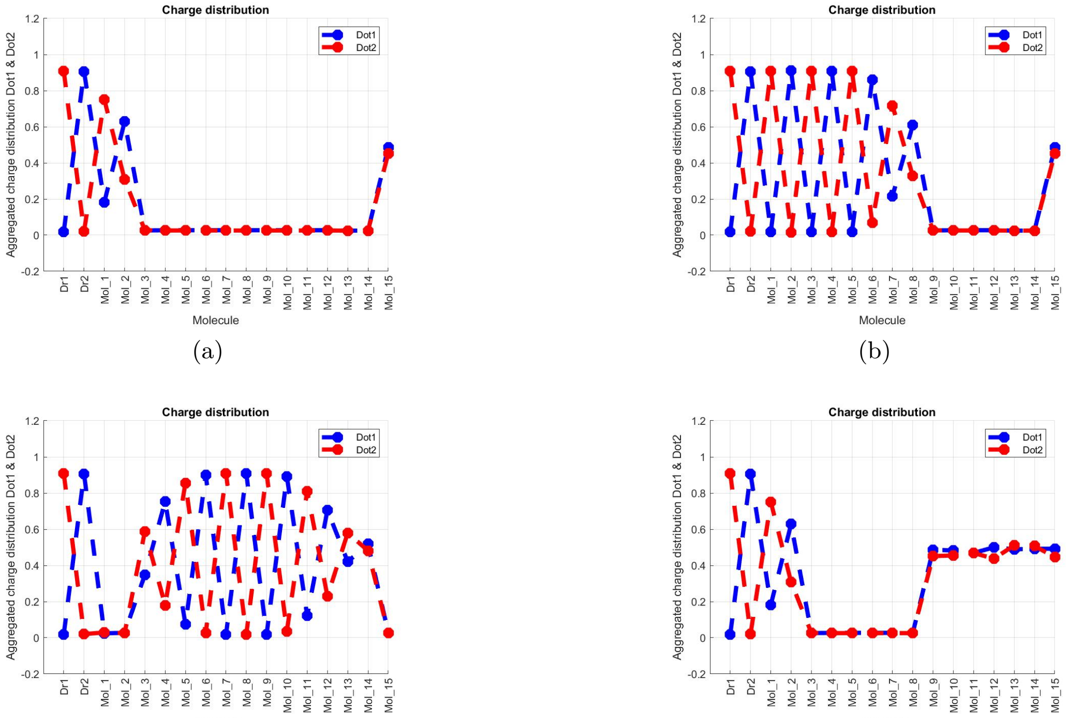
Fig. 5.8: Output voltages and charge distribution on an eight molecules wire subjected to an input voltage $V_{in} = 0.4V$ and $V_{in} = 0.5$V

Similar considerations can be carried out for longer wires. Before moving on, notice that only two clock regions have been used. This solution can be critical in some cases. Indeed, for too-long wires, the molecules farther to the driver will tend to distribute their charges on the dots before the arrival of the authentic information. This behavior creates the possibility of clash events. Moreover, having just two clock regions means creating a direct connection within the central cells of the different neurons. Theoretically, this is not a problem if the wires length is the same for the three interfaces. However, it is not always possible to ensure this condition, so it is better to introduce a third clock region splitting the output wire. In this way, it is possible to achieve adiabatic propagation.
Another critical consideration concerns the dimension of each of the two clock regions in the output wire. Take as an example a molecular wire organized in the following way:

- the first two molecules represent the central cell of a first neuron and are placed in the first clock region
- output wire made of twelve bis-ferrocene molecules organized as follows:
  - eight molecules in the second clock region
  - four molecules in the third clock region
- two interface cells presented at the end of the output wire and influenced by the third clock signal, making the total number of molecules in this region equal to six. The saturation voltage of the molecules in this cell is equal to 1V.
- two final bis-ferrocene molecules forming the central cell of a second neuron

This simulation is interesting since it contains almost all the elements that will be present in the final solution for the output chain. Moreover, looking at the results reported in figure 5.9, it is possible to derive some critical points related to the clock regions organization. For this example has been applied $V_{in} = 0.65V$ to make the wire saturates. Due to the saturation of the wire, the propagation of the information starts to seem more compliant with adiabatic propagation theory. However, the main problem is that as soon as the first clock region is deactivated, the molecules in the second one lose the charge distribution, and the information is completely lost. So it is possible to say that even with the proper conditions in terms of clock regions and saturation, the transport of digital information can have some serious problems. Therefore, it is needed to increase the length of the whole wire or, at least, the number of molecules present in the second clock region. As an example, consider the results reported in figures 5.10a-5.10b, there is considered an eighteen molecules wire with the same clock region division seen in the previous case. The number of molecules present in the second zone is increased to eight. From these last results, it is even more clear that the digital solution for propagating the information through the neural network is the best to provide a stable and reliable solution to the problem. However, to achieve it, the design must respect the constraints related to the number of molecules and the voltages applied to the structure. The first one is achievable by correctly placing molecules in the different clock regions. The second is instead more critical. In the previous chapters, was showed that the voltage at the neuron output could be minimal depending on the input values. If the 1nm solution is chosen, the propagation cannot occur since the wire is not saturated, and the information is lost. Moreover, there is a second problem to be considered. In Chapter 4 was performed a parametric analysis of the weights concerning intermolecular distance variation: with increasing distance, the weights would reduce almost exponentially. That result is confirmed in figure 5.10b.
The wire has to be implemented with a valuable amount of molecules distributed in the clock regions to provide correct adiabatic propagation. Even though the information arrives at the wire end and, therefore, at the interface cell, the consequent polarization of the central cell is extremely low. This is a significant problem whose solution can be just the reduction of the intermolecular distance.
These results are also confirmed by the voltage analysis reported in figure 5.11 and become even more evident by taking into account molecules with α higher than 1V. At the end of

this part of the discussion, it is again apparent how the digital solution should be preferred to the analog, even if it will imply the loss of the exact value computed at the neuron site.



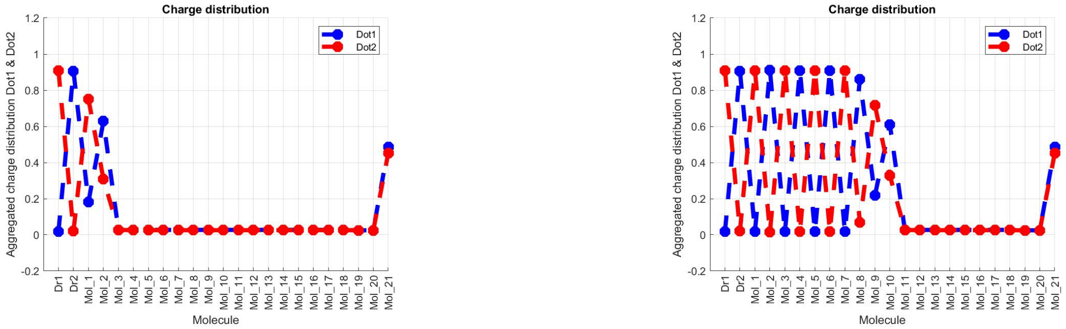
(a)

(b)

(c)

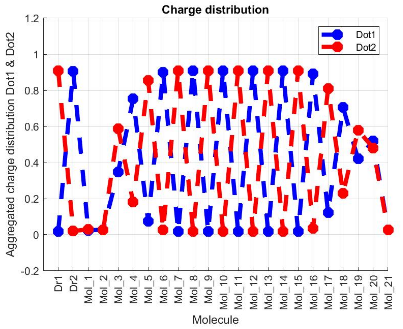
(d)

Fig. 5.9: Propagation on the complete wire and connection with the second neuron central cell: Vin = 0.65V, 12 molecules on the wire, 2 interface molecules and two bis-ferrocenes molecules implementing the connection with a second neuron



(a)

(b)



(a)

(b)

Fig. 5.10: Charge distribution plot of the propagation across an eighteen molecules wire and connection with a second neuron across an interface cell $\alpha = 1V$

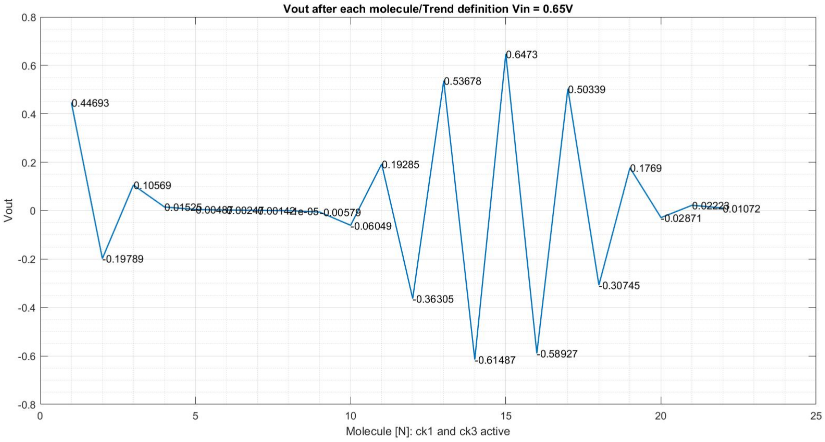


Fig. 5.11: Voltage trend analysis across a N = 18 molecular wire with interface cell at the end. Voltages at the end of the molecular chain are extremely low.

5.2 Clock profile definition

The critical points found in the previous sections are related to how to saturate a molecular wire characterized by d = 1nm. The analog solution with the structures adopted so far seems to be not a concrete possibility. Indeed, there are two main problems: the requirement of a minimum number of molecules in the second clock region and the information fading with distance. A possible idea to overcome these problems and achieve the analog implementation of the final neural network is to introduce step variations in the clock profile. These are already present in FCN solutions given the impossibility of generating a square wave with perfect vertical edges.

As a consequence, the clock signal is going to take different intermediate values between the extremes -2$\frac{V}{nm}$ and +2$\frac{V}{nm}$.
In a first approximation, the analyzed clock signals are those reported in figure 5.12. For simplicity and preliminary evaluation of possible achievable behavior, the involved wire divides into just two clock regions.

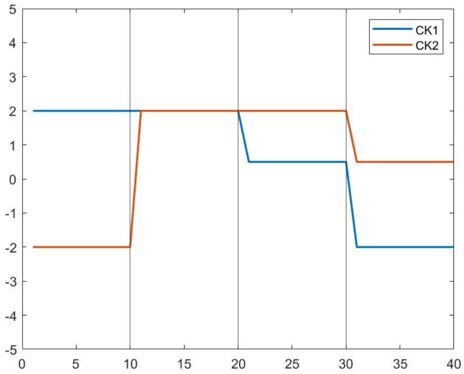
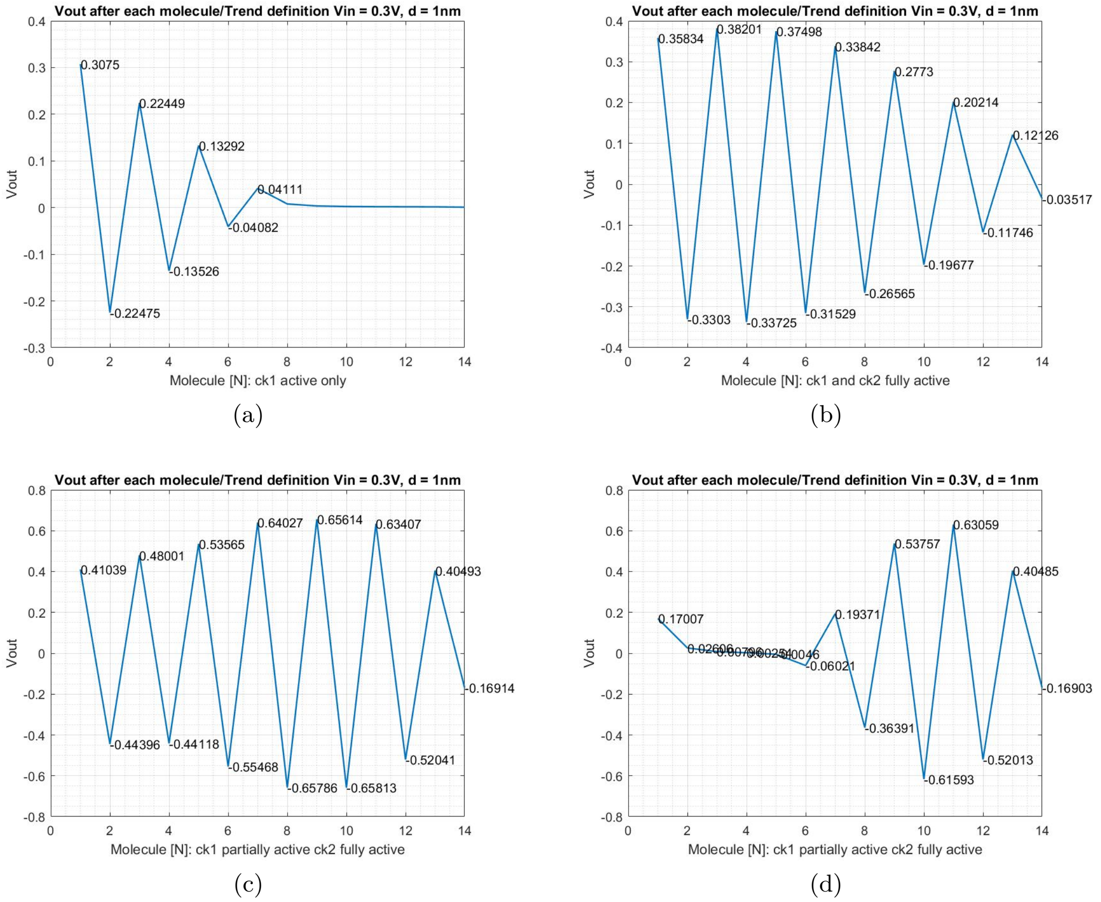


Fig. 5.12: First ideal clock signals

It is important to remark that the proposed profile is just a first approximation to move out from ideality. The step variation is present only on the falling edge of each signal, which have a shift in phase one to the other. The outcomes of the first simulation are reported in figure 5.13. The input voltage is 0.3V, and the number of molecules present in the wire equals 14. The molecules are all bis-ferrocenes. Notice that with the ideal clock configuration, the consequences of border effects would be evident and, therefore, a substantial reduction of the voltage at the end of the wire. In the results reported in 5.13 it is evident that thanks to the presence of the slower variation of the clock signals, the propagation is sustained, and the molecules in the wire have maximum charge separation. It is interesting to notice that the propagation properly occurs in this case. Due to the usual reasons, some border effects are still present on the last molecule. In figure 5.13d it is important to notice that, even if the first clock region in the wire is turned off, the molecules in the second one are capable of maintaining their charge distribution. This is the main effect of introducing the intermediate values in the clock. Moreover, comparing this result with the one reported in figure 5.13b, it is possible to notice a reduction in the border effect.
From this first analysis, the idea of avoiding turning off immediately the first clock signal seems to be engaging to maintain and stabilize the information along the wire. Different simulations were performed involving different wire lengths. However, remarkable differences with the case reported cannot be highlighted.



(a)

(b)

(c)

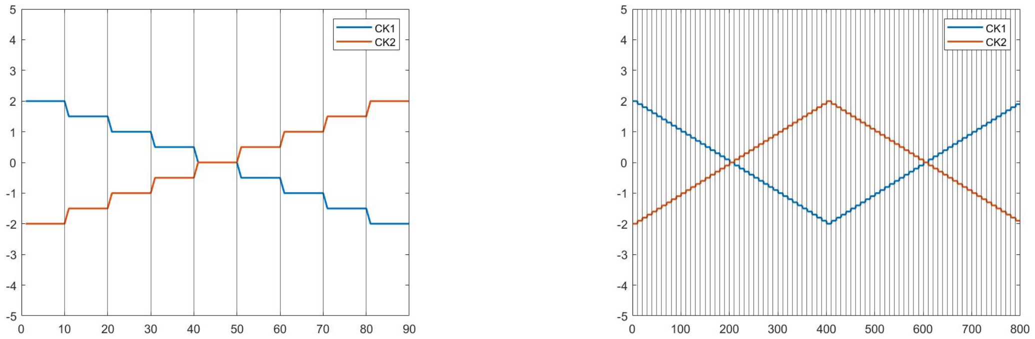
(d)

Fig. 5.13: $V_{in} = 0.3V$, N = 14, d = 1nm

5.2.1 More realistic clock profile

As already said, the clock signal waveforms introduced in the previous section are highly idealized. It is important to understand if the benefits pointed out so far can also be confirmed with the introduction of a more realistic clock, in which a finite number of steps characterizes both the falling and rising edges of the clock signals. The updated clock signals are reported in the graphs 5.14a and 5.14b.

The two figures differ mainly in the number of steps introduced in the transition. Indeed, the waveforms in 5.14a are characterized by steps of height $0.5\frac{V}{nm}$, while in the other case those are equal to $0.1\frac{V}{nm}$. The results found in the two situations can be considered comparable; for this reason, to improve the computational time of each simulation, only the results related to the use of the first signal will be presented. An important aspect to be noticed is the scale of the x-axis of the plots. These show the steps multiplied by 10 for plotting easiness. The simulation results proposed in this section are referred to by the step number, from which the values of the two clock signals at that time instant can



(a)

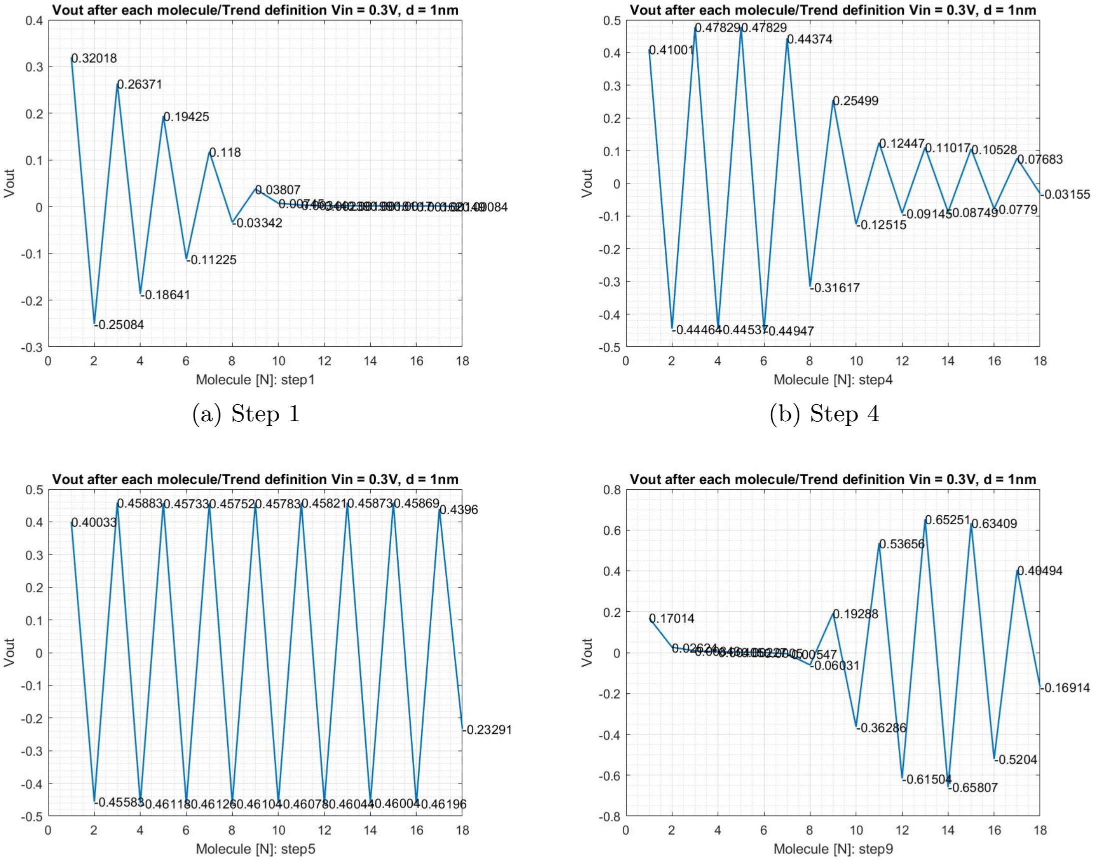
(b)

Fig. 5.14: Complete clock waveforms

be derived.
The first proposed simulation has the following characteristics:

- N = 18 bis-ferrocene molecules
- intermolecular distance equal to 1nm
- $V_{in} = 0.3V$. Notice that with clock signal showing vertical transitions, such a value would not provide the saturation of the molecules in the wire.

The results are reported in figures 5.15a-5.15d. The first plot in figure 5.15a represents the first time instant, in which only the clock region number one is active. On the last molecule there is a strong influence of the border effects. In 5.15b the second clock region activates, and its effects are visible on the voltages present in the first zone. The wire then saturates when all the molecules are subjected to a clock field equal to $0\frac{V}{nm}$, whose plot, in terms of charge distribution across the dots, is reported in figure 5.16. It is interesting to notice a lower charge separation due to the reduced value of the electric field at that time step. Then, in figure 5.15d is reported the last propagation step, in which it is possible to appreciate a useful polarization on the last cell of the molecular wire.



(a) Step 1

(b) Step 4

(c) Step 5

(d) Step 9

Fig. 5.15: $V_{in} = 0.3V$, d = 1nm, N = 18

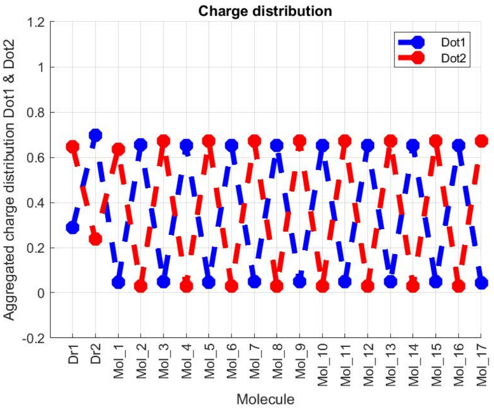


Fig. 5.16: First simulation: charge distribution for CK1 = CK2 = $0\frac{V}{nm}$

Overall, it is possible to consider these first results satisfactory. We tested different input voltage values, such as 0.1V, obtaining the same results as those reported above. This implies that the behavior of a wire clocked in this way is independent of the input voltage level. Once again, the final result is a binary wire. Therefore, it was not possible to obtain the analog solution for any input.

Another critical value is the polarization of the last cell of the wire, being this the one in contact with the linked neuron. Also, in this case, the values are almost equal independently of the number of molecules present in the molecular chain.

Summing up the results presented, it is possible to state that the best solution for information propagation is a digital one. The slow variation of the clock allows reaching the saturation for any values at the input with an intermolecular distance equal to 1nm.

5.3 Saturator molecules

It is important to summarize all the results found in the analysis related to the output wire. A list of them is reported here below.

- **Number of molecules**: the number of molecules, after a certain threshold value, does not change the wire behavior. A low amount in the second region can lead to the information loss as soon as the first clock region is turned off. This last observation is true in both the cases of ideal and non-ideal clock signals.
- **Type of molecules**: the only useful molecule to implement the molecular wire is bis-ferrocene. If molecules with higher V_{sat} are involved, the result is the complete loss of the information.
- **Intermolecular distance**: the intermolecular distance at this point has to be selected. The starting idea was to implement analog propagation in the circuit. However, the results show that the digital implementation fits better the requirement of having good propagation and a useful voltage at the end of the wire. Moreover, the weights associated with a charge separation equal to 1nm are too low for successful neuromorphic computation at the neuron site.

The last analysis consists of the introduction of saturator molecules. These are ad-hoc characterized molecules whose saturation voltage is lower than the one of bis-ferrocene. In figure 5.17 is reported the VACT for a 0.1V saturator molecule considered with a clock signal equal to $+2\frac{V}{nm}$. This molecule, positioned either at the beginning or at the end of the molecular wire, ensures that saturation is reached for each input voltage and sustained until the end with d = 1nm. In other words, it has to guarantee a digital solution for any case when the intermolecular distance is high.

The behavior of this molecule, when placed at the beginning of the wire, is evident and will be analyzed in great detail in the following sections. It is worth providing some results related to its possible influence at the wire end.

Consider the case in which an eighteen molecular wire is subjected to an input voltage equal to 0.3V. The last cell of the wire is made by $\alpha = 0.1V$ molecules. The results are reported in figures 5.18a-5.18b, representing the first and the last steps of the propagation.

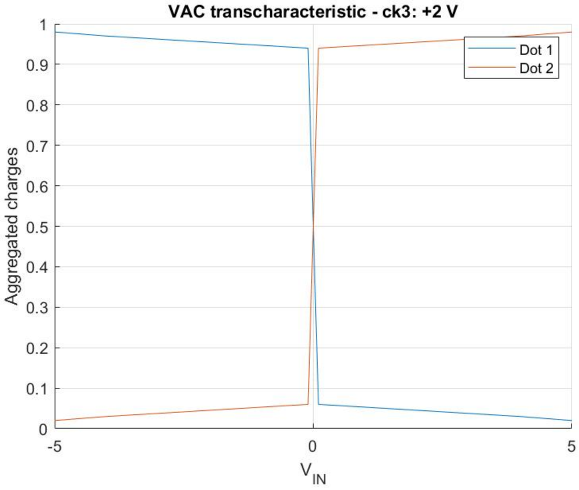
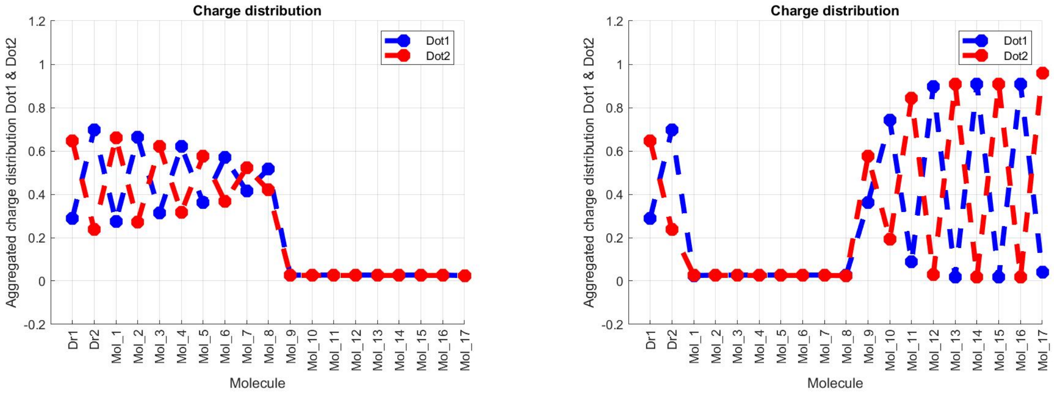


Fig. 5.17: VACT $\alpha = 0.1V$, clk $= 2\frac{V}{nm}$



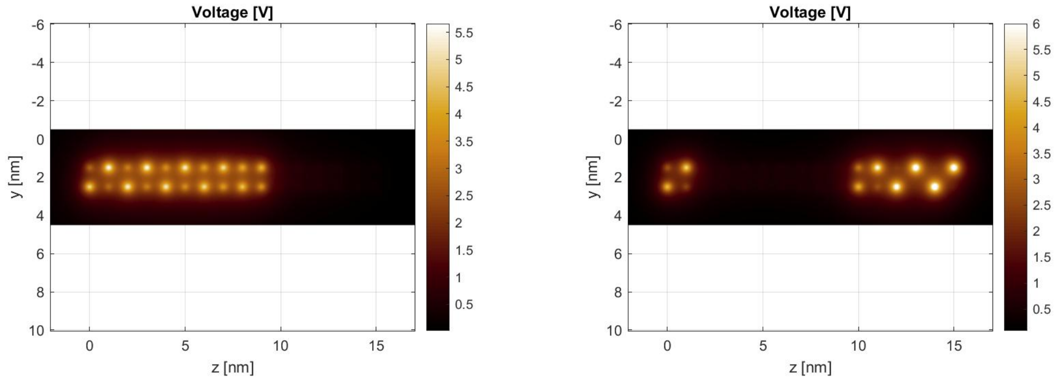
(a) Step 1

(b) Step 9

Fig. 5.18: $V_{in} = 0.3V$, d = 1nm, N = 18. Saturator molecules at the end of the wire

These results have to be compared with those presented in figure 5.8b, even though there is no presence of interface molecules at the end of the wire. Besides that, it is evident how the saturator increases the polarization at the end of the wire, whose value is equal to -0.9084. Consequently, the related interface influence on the connected wire is more significant. This opens the possibility of using the saturator molecules to increase the voltage level at a specific interface, making it more influent than the others connected to the same neuron.

As a second possible use of the saturator molecules, it is worth to mention the realization of smaller clock regions. Indeed, it is possible to demonstrate that, without the saturators, the minimum number of molecules that have to be present in the second clock zone is equal to six with d equal to 1nm. In figure 5.19 are shown the results for a molecular wire made of fourteen bis-ferrocenes, in which six of them are localized in the second clock region. Therefore, it is possible to state that the saturator molecules at the end of the wire are going to make possible the realization of more stable digital local interconnections. The value of the polarization on the last cell is -0.9084.



(a) Step 1

(b) Step 9

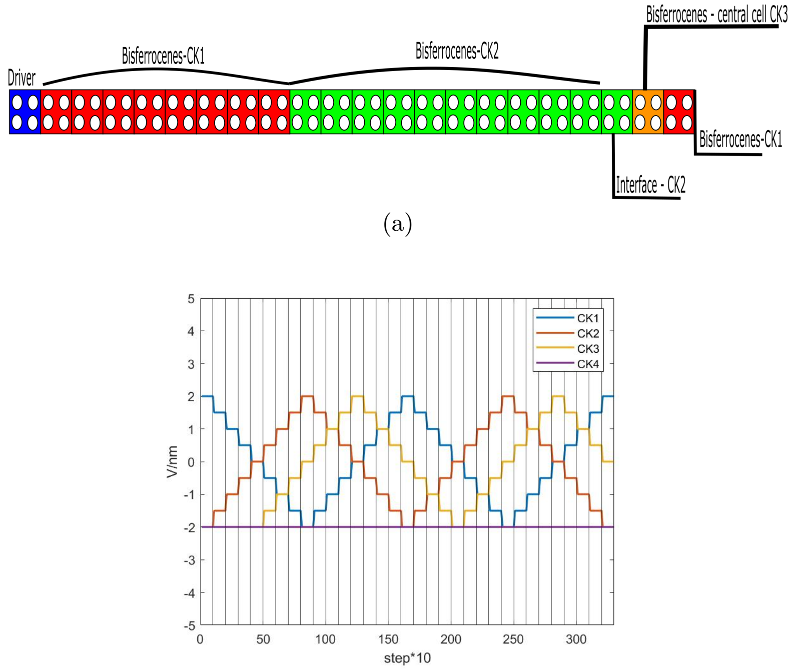
Fig. 5.19: $V_{in} = 0.3V$, d = 1nm, N = 14. Saturator molecules at the end of the wire

Through this study, it was proven that that the presence of molecules with a steep VACT curve can provide some improvements in the digital propagation of the information along a molecular wire. In the following of this work, saturator molecules will be mainly proposed at the beginning of the wire, unregarding the intermolecular distance.

5.4 Final choice for *d* and connection with the second neuron

In the previous sections, were delivered several ground considerations to understand the final structure of the axon of the molecular neural network. Different techniques were presented to work with an intermolecular distance equal to 1nm. The analog propagation of the information was not achieved, and the wire had to be saturated, implementing a binary network. The remaining problem related to a high separation is the one concerning the low weights of the interfaces. For this reason, provided that the final neural network will be a fully digital one, it is better to work with 0.9nm within the molecules. This way, saturation is ensured along the wire, and the weight coefficients have proper values for neuromorphic purposes.

It is also possible to complicate the wire structure to analyze better the different components' behavior in a whole system. The design of the adopted molecular wire is reported in figure 5.20a. There are three clock regions. The first two split the propagation along the wire according to the analysis performed in the previous sections, while the third clock influences only the bis-ferrocene cell connected at the end of the wire. The interface cell is localized in the second zone. Moreover, a further cell is added at the far end to simulate propagation after the computation at the neuron site. It is possible to appreciate the waveforms and time relations of these signals in figure 5.20b.



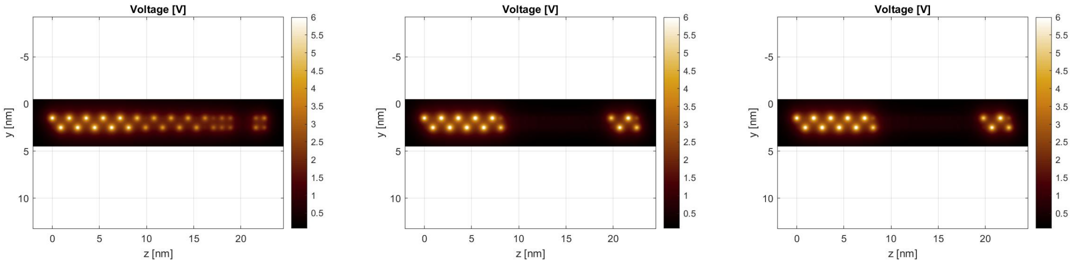
(a)

(b)

Fig. 5.20: Clock regions organization and timing. Preliminary assumptions

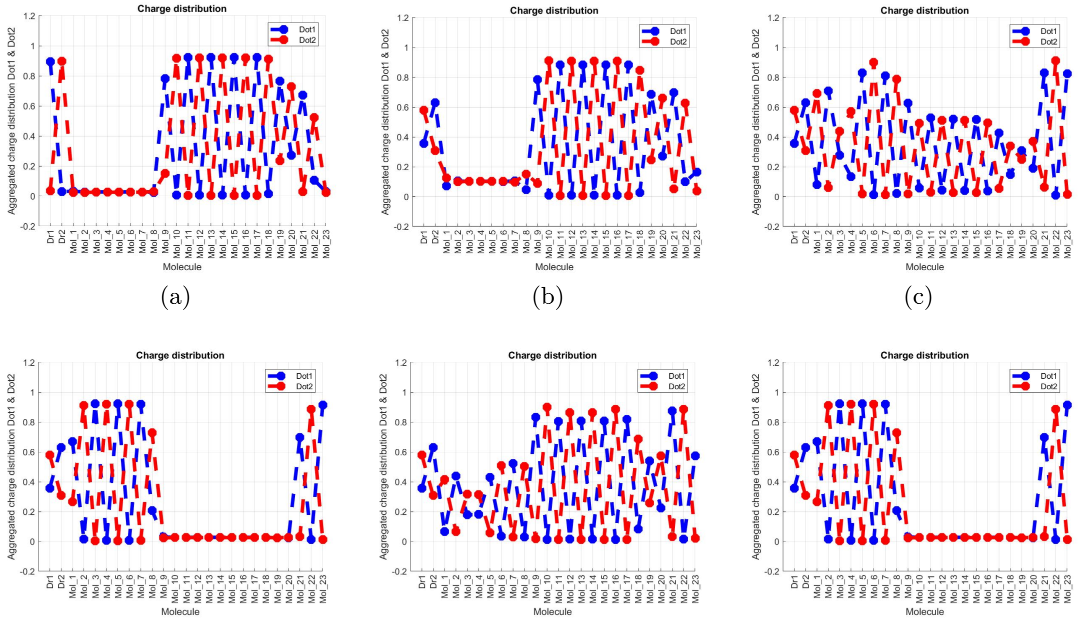
Notice how the three regions behave: CK1 and CK2 are in complete phase rotation, i.e,. the first's maximum value corresponds to the second's minimum. The third region takes the value present in the second region and propagates it to the first one again, so to the last cell of the wire. Focusing the attention on the propagation behavior, it is possible to study the results from a simulation in which the input voltage was set to be 0.65V and the distance to 0.9nm. These are reported in figures 5.21a-5.21c. In 5.21a, it is possible to notice the moment in which the information reaches the bis-ferrocene cell. Consider that the last cell, governed by the first clock signal, presents some small charge distribution. The information propagates through the wire and arrives at the interface cell, which is characterized by $\alpha = 1V$. The logic value is correctly evaluated after the interface, and the propagation can proceed. From a timing point of view, this simulation confirms the correctness of the adopted clock distribution. However, some problems can be highlighted if the goal is to transport multiple successive binary values. This can be easily understood from a practical example, reported in figures 5.22a-5.22f.

In this last case, the objective was to correctly transport two different binary values with corresponding input values from the drivers equal to -0.6V and 0.2V. Bis-ferrocene molecules form the wire, and the interface has $\alpha = 1V$ molecules. As in the previous example, the neuron central cell and the output molecules are connected at the wire end.



(a) (b) (c)

Fig. 5.21: Potential plots: propagation along the complete molecular wire. $V_{in} = 0.65V$, d = 0.9nm



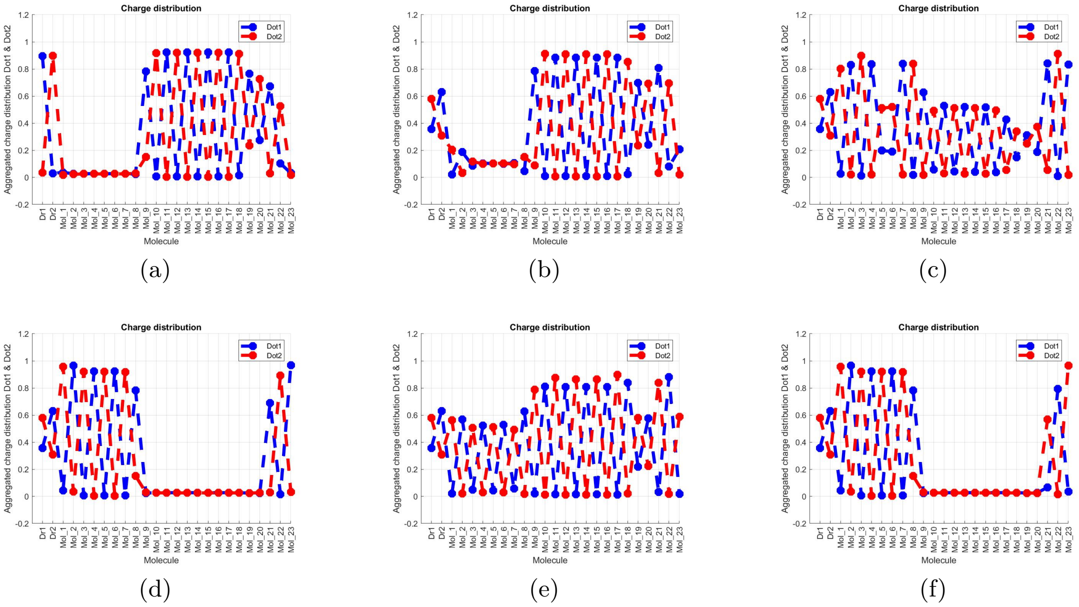
(a) (b) (c)

(d) (e) (f)

Fig. 5.22: Multiple inputs propagation. The clash event is evident from the charge distribution plots

The first snapshot is related to propagating the first logic value; in the second one, the input changes, and the corresponding logic value starts to move. The main problems are evident in 5.22c in which a clash occurs, making the propagation through the wire extremely unstable with the possibility of complete loss of the correct information.
This behavior derives from the back-propagation from the second clock region while the first is trying to transport an opposite binary value. A possible solution is to introduce the saturator molecules at the beginning of each wire to solve this problem. In this way,

the strong polarization obtained in the saturation cell will overcome the effects of back-propagation and the new charge configuration. In figures 5.23a-5.23f are reported the results of the simulation. The other parameters remain unchanged.



(a)

(b)

(c)

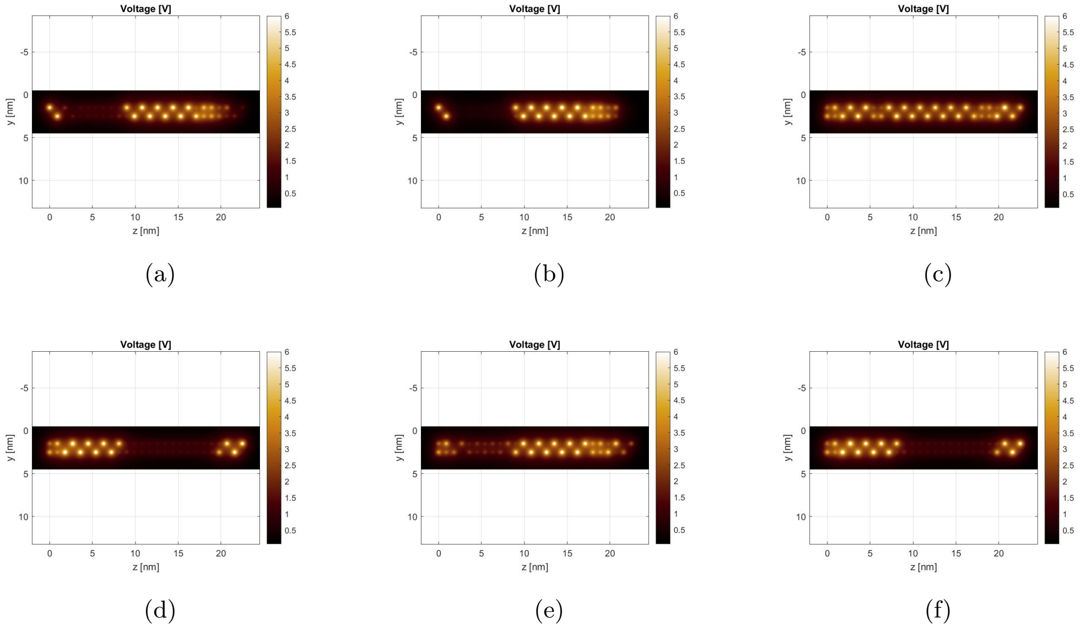
(d)

(e)

(f)

Fig. 5.23: Multiple inputs propagation with the introduction of saturator molecules

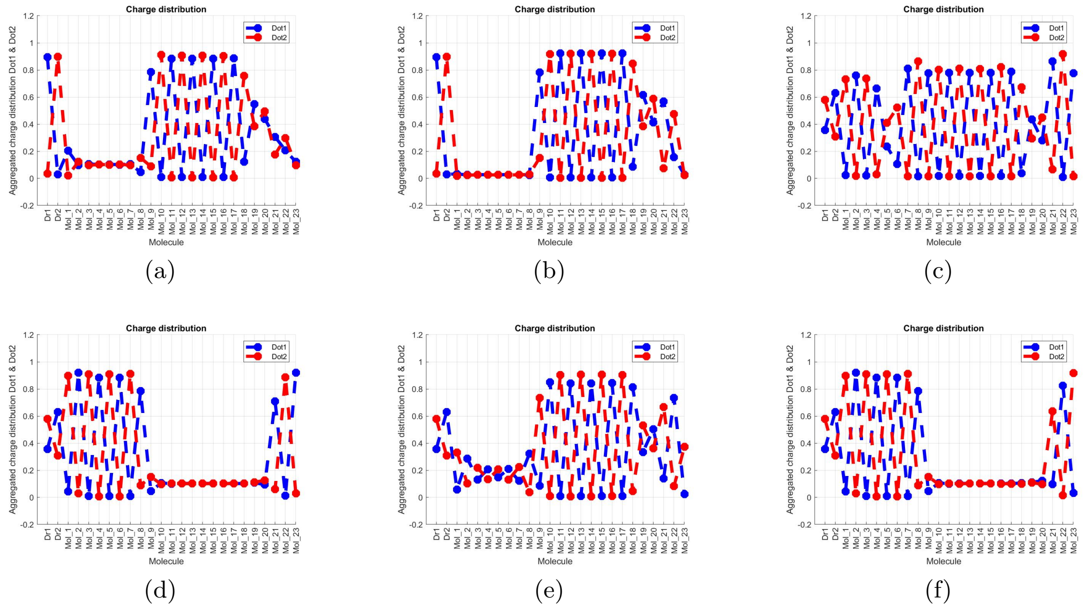
The issue seems to be solved. However, the introduction of molecules with such a small saturation voltage will lead to important issues if an interface with a high α parameter is present. Indeed, as soon as the first clock zone gets active, the saturator molecules polarize randomly in one of the two stable configurations, biasing their charges depending on spurious electric field influences. This effect combines with the low polarization of interface cells with high saturation voltage. The final result is that the information could be completely lost also in this case since the system is not reliable. Simulation results are proposed in figures 5.24a-5.24f. The idea of saturating the wire at the front end is positive and is not mandatory to renounce to it. Indeed, it can be shown that by increasing V_{sat} of the saturator up to 0.3V, thus maintaining it lower than 0.5V, it is possible to save the benefits related to the use of such molecules and, at the same time, avoid information loss. The results showing this concept are reported in the pictures 5.25a-5.25f. In figure 5.26 is reported the VACT of the new saturator molecules.



(a) (b) (c)

(d) (e) (f)

Fig. 5.24: Multiple input propagation with interface cell having $\alpha = 1.5V$



(a) (b) (c)

(d) (e) (f)

Fig. 5.25: Multiple input propagation with interface cell having $\alpha = 1.5V$ and $\alpha = 0.3V$ on the saturator cells

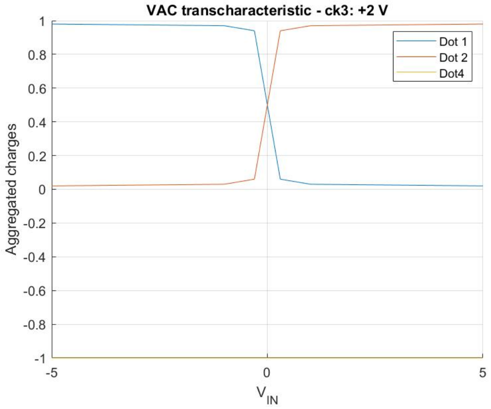


Fig. 5.26: VACT $\alpha = 0.3V$, clk $= 2\frac{V}{nm}$

The only problem remaining is the possible presence of clash events in propagation. Even though the introduction of low V_{sat} molecules allow for the correct value reconstruction; this is an unstable situation that is better to avoid. Moreover, the sudden configuration change within a molecule would increase the overall power dissipated by the structure. The main reason this situation can occur is, as already explained, the back-propagation from the second region. It is due to the absence of a reset state in the clock configuration adopted so far. For this reason, the clock signals must be modified. The change has to combine with the saturator molecules' effects, providing a stable and safe transport of the information through the entire molecular wire for every interface.

5.5 Clock signal change and *three-wires* layout analysis

As anticipated at the end of the previous section, the clock signal waveforms have to be modified to avoid problems related to clashes. The waveforms are reported graphically in figure 5.27. It is essential to highlight the differences from those used in the preceding simulations. First, in this case, each clock signal has fewer steps on its edges. Specifically only one is present at $0\frac{V}{nm}$. That allows for the reduction of the computational cost per simulation.
The second significant variation is the introduction of a fourth clock signal. It is mainly needed to isolate the saturator molecules from the rest of the circuit, activating them only when the computation is stable on the central cell of the neuron. Without this trick, the risk is to have back-propagation from the saturator molecule to the interface cells, affecting the correct evaluation of the final output. Also related to this signal, notice how it is time-shifted to the others. This solution allows to take the logic value from the central cell and transport it to the output wire. Without this shift in time, the risk is that, for low polarization values on the central cell, the charge configuration goes down to zero to move out from the unstable situation.

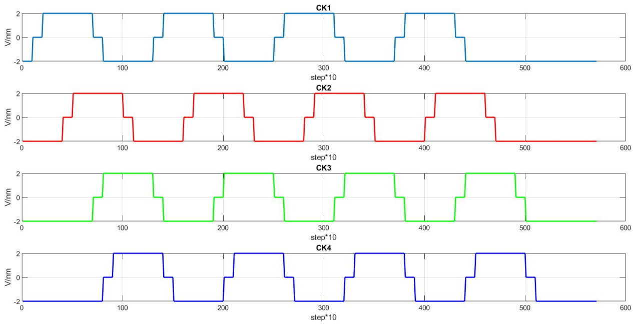
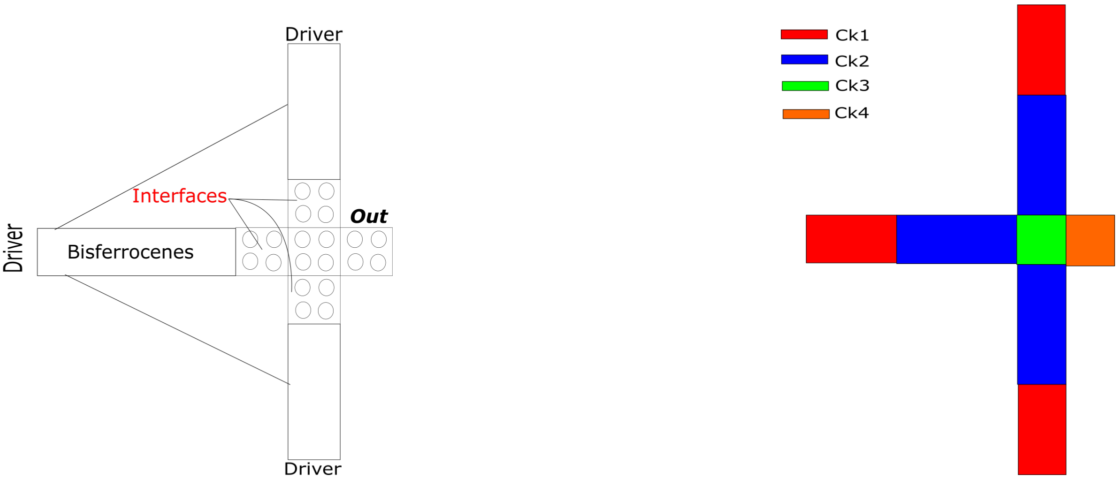


Fig. 5.27: New clock signals

This system of signals was proven to work correctly with the single wire cases, and, therefore, it has been possible to test it on a complete structure. In this case, all interfaces are present, and the inputs come from three input wires. The schematic representation of the structure is reported in figure 5.28a, while in 5.28b is presented the clock region distribution.



(a)

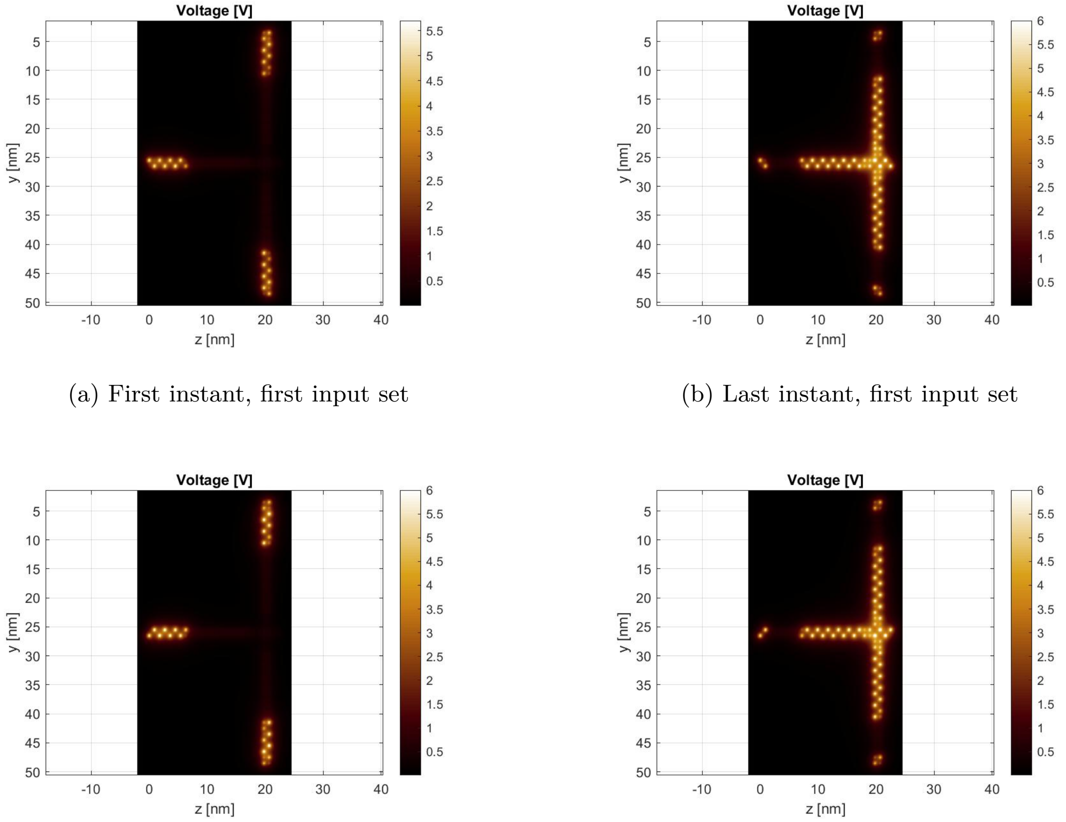
(b)

Fig. 5.28: Circuit schematic and clock regions organization

As a first simulation in order to prove the validity of the developed structure consider the following, in which multiple inputs propagation is desired:

- m1: $\alpha = 1V$, V_{in} = [0.3V, 0.3V]
- m2: $\alpha = 2V$, V_{in} = [-0.5V, 0.8V]
- m3: $\alpha = 1.5V$, V_{in} = [-0.5V, 0.8V]

It is possible to appreciate the results of this simulation in figures 5.29, in which are reported the initial and final time instants for each of the two proposed input sets.



(a) First instant, first input set

(b) Last instant, first input set

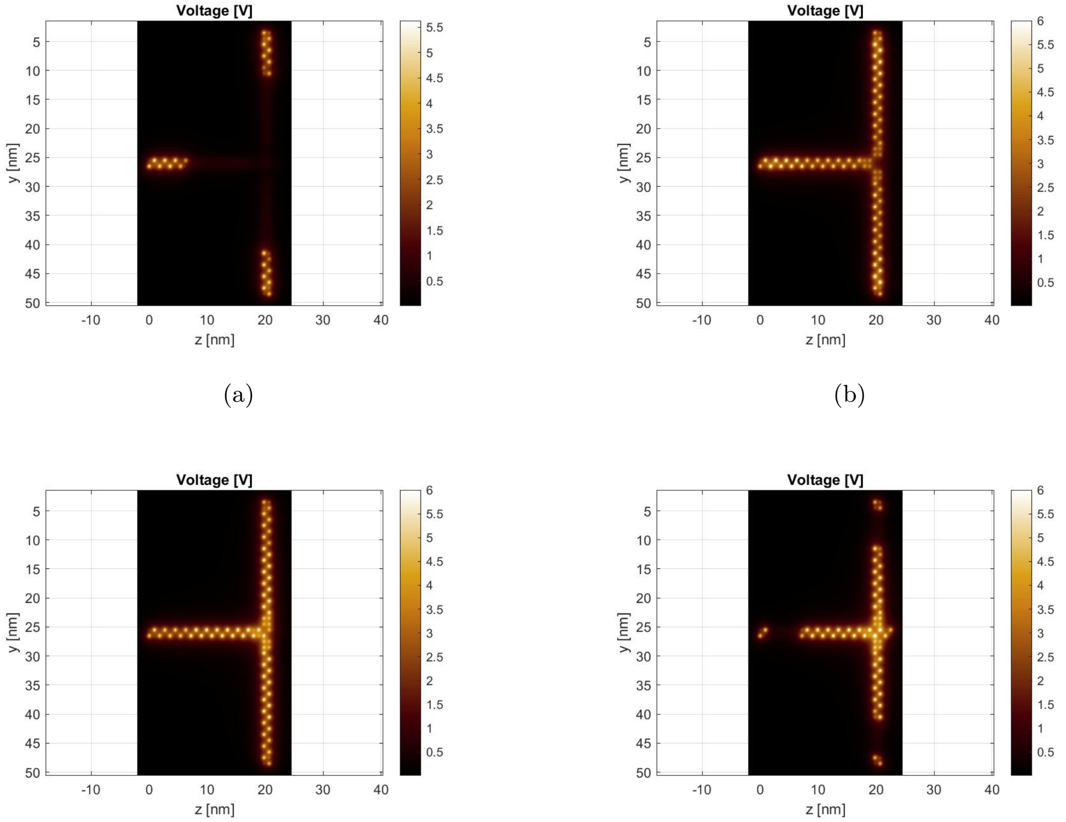
(c) First instant, second input set

(d) Last instant, second input set

Fig. 5.29: Simulation results for the first proposed simulation

It is important to highlight that the propagation of the binary information through the structure was successful in both cases. The adopted circuit solution is working correctly, and the effects related to back-propagation and clash events during the input change have been completely removed. Another critical study concerns the possibility of reversing the usual behavior of the majority voter, already implemented for both *drivers only* and *additional cell* layouts. To have such behavior, consider the inputs configuration suggested here below, and whose results are reported in figure 5.30.

- m1: $\alpha = 2V$, V_{in} = -0.3V
- m2: $\alpha = 2V$, V_{in} = -0.4V
- m3: $\alpha = 1V$, V_{in} = 1V



(a)

(b)

(c)

(d)

Fig. 5.30: Simulation results from the second proposed simulation

This simulation proves that it is possible to achieve the inversion of the usual behavior of the majority voter with the *three-wires* circuit. This result has been obtained by introducing two interfaces presenting a high saturation voltage while the other is characterized by $\alpha = 1V$, thus having the maximum possible weight.
As a final remark, notice that the exact input values present at the drivers just indicate the binary values that will be propagated along the corresponding wire. A positive input value will correspond to a logic '0', while a negative voltage to a logic '1'. At the interface inputs, the voltage levels will be around 0.5V, considering the influence of saturation and border effects.

5.5.1 Voltage analysis

Even though the neural network analysis is moving toward a logical implementation, it is useful to continue the study of the analog values that can be obtained in different working situations.

The involved circuit is the one presented in the previous section. For what concern the analysis is essential to remind the explanation about figure 4.25 in Chapter 4. The different effects presented in that section will also be important in this case and, for some input combinations, will be effective in the final computation. A comparison is going to be provided between the results obtained with *additional cell* layout and the *three-wires* structure. Such an analysis is important to understand how the saturation of the molecules in the wire and the consequent loss of the starting analog value affect the computation. The coefficients introduced in the weighted input formula are the ones presented in table 4.4. As a last remark, it is essential to fix how the sign of the voltage values has been considered according to the logic value propagating along the corresponding wire. Look at the schematic presented in figure 5.31. There are reported voltage values considered for the computation in the circuit characterized by the input wires. Notice that a simplification was applied: the actual sign of the voltage at the input of a particular interface is considered negative if the logic value is '1' and positive if '0'. This assumption considers the voltage across the two molecules of an interface cell to be equal, which is not always correct due to the diagonal coupling. However, the results obtained with this assumption can be considered successful and generally in line with the expectations.

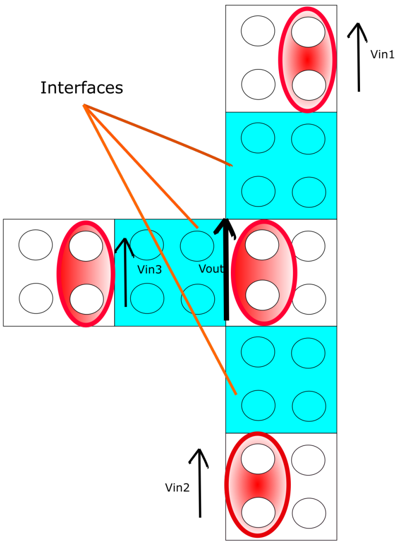


Fig. 5.31: Schematic of the chosen voltages to perform computation on the neuron

The results of the different simulations are reported in the following way:

- For each input and interface combination, were performed the simulations with *additional cell* and *three-wires* layouts
- For the *additional cell* were computed the ideal results
- Related to the *three-wires* are presented three different results:

- $Vout_a$: output voltage prediction before the activation of the central cell. So the input values are evaluated before the activation of the central cell and according with the schematic in figure 5.29
- $Vout_b$: output voltage prediction with input values evaluated when the central cell is active, these are influenced by back-propagation
- $Vout_c$: actual value taken from the outcome of the simulation

The numerical results are reported in tables 5.1 and 5.2. There are four possible types of results, indicated with different colors in the tables. The first one is the correct and normal behavior of the neuron, which behaves as a majority voter and whose outcome is in line with the numerical evaluation of the neuron formula 3.2. The yellow rows concern those simulations in which the input having the higher weight propagates to the neuron output besides being the less recurrent. These results can be correctly predicted. The errors, so the situations when there is an opposite polarity between the simulation and prediction voltages, are highlighted in red. At last, in green, a situation in which the results provided by the additional cell structure and the complete one have different polarity is reported. The most important consideration about these results is that, for both layouts, it is possible to predict the voltage sign on the first molecule in the central cell with a high correctness rate. This is valid for the case of normal or reversed behavior of the majority voter circuit. It is worth mentioning that the differences between the ideal and actual values are lower in the *additional cell* neuron structure. This result is clearly due to the induced saturation characterizing the wires of the more complex system. For what concern the presence of prediction errors, a more in-depth analysis is required.

Output voltage results: additional cell neuron					
Nr.	**m1**	**m2**	**m3**	V_{out} **ideal**	V_{out} **real**
1	$\alpha = 1.5V$ Vin = -0.3V	$\alpha = 1V$ Vin = 0.4V	$\alpha = 2.5V$ Vin = -0.5V	0.088V	0.11364V
2	$\alpha = 2V$ Vin = 0.5V	$\alpha = 2V$ Vin = 0.4V	$\alpha = 2V$ Vin = -0.3V	0.094V	0.02354V
3	$\alpha = 1.5V$ Vin = -0.3V	$\alpha = 1.5V$ Vin = 0.4V	$\alpha = 1.5V$ Vin = -0.5V	-0.03338V	-0.0295V
4	$\alpha = 2V$ Vin = 0.6V	$\alpha = 2V$ Vin = 0.4V	$\alpha = 1V$ Vin = -0.5V	-0.115V	-0.07893V
5	$\alpha = 1V$ Vin = 0.4V	$\alpha = 1V$ Vin = 0.4V	$\alpha = 1.5V$ Vin = 0.5V	0.317V	0.376V
6	$\alpha = 1.5V$ Vin = -0.5V	$\alpha = 1V$ Vin = -0.4V	$\alpha = 1V$ Vin = -0.4V	-0.317V	-0.44979V
7	$\alpha = 1.5V$ Vin = -0.2V	$\alpha = 1V$ Vin = 0.3V	$\alpha = 2V$ Vin = -0.6V	0.06065V	0.09278V
8	$\alpha = 1.5V$ Vin = -0.2V	$\alpha = 1.5V$ Vin = 0.4V	$\alpha = 2.5V$ Vin = -0.8V	0.02254V	0.01653V
9	$\alpha = 2V$ Vin = -0.3V	$\alpha = 2V$ Vin = 0.4V	$\alpha = 2.5V$ Vin = -0.5V	0.04103V	0.03168V
10	$\alpha = 2V$ Vin = 0.3V	$\alpha = 2V$ Vin = -0.7V	$\alpha = 2.5V$ Vin = 0.5V	-0.04103V	-0.01974V
11	$\alpha = 1V$ Vin = 0.3V	$\alpha = 1V$ Vin = -0.7V	$\alpha = 2.5V$ Vin = 0.5V	-0.127V	-0.10374V
11	$\alpha = 1V$ Vin = 0.3V	$\alpha = 1V$ Vin = -0.7V	$\alpha = 1V$ Vin = 0.5V	0.086V	0.22863V
13	$\alpha = 1.5V$ Vin = -0.5V	$\alpha = 2V$ Vin = 0.5V	$\alpha = 2.5V$ Vin = -0.5V	-0.025V	-0.04342V
14	$\alpha = 1.5V$ Vin = 0.3V	$\alpha = 2V$ Vin = -0.3V	$\alpha = 2.5V$ Vin = 0.3V	0.01521V	0.0.00014V
15	$\alpha = 2.5V$ Vin = -0.5V	$\alpha = 1.5V$ Vin = 0.4V	$\alpha = 2V$ Vin = -0.3V	0.0127V	0.03647V
16	$\alpha = 2.5V$ Vin = -0.5V	$\alpha = 2V$ Vin = 0.4V	$\alpha = 1V$ Vin = -0.3V	-0.116V	-0.08118V
17	$\alpha = 1.5V$ Vin = -0.3V	$\alpha = 2.5V$ Vin = 0.5V	$\alpha = 1.5V$ Vin = -0.5V	-0.03613V	-0.07495V
18	$\alpha = 1.5V$ Vin = 0.4V	$\alpha = 1.5V$ Vin = -0.2V	$\alpha = 2.5V$ Vin = 0.8V	0.04054V	0.03979V

Table 5.1: Output voltages simple majority voter

Output voltage results:neuron with input wires			
Simulation number	$Vout_a$	$Vout_b$	$Vout_c$
1	0.157V	0.195V	0.39918V
2	0.157V	0.195V	0.39918V
3	-0.045V	-0.037V	-0.28521V
4	-0.151V	-0.174V	-0.42176V
5	0.698V	0.534V	0.52035V
6	-0.447V	-0.406V	-0.5136V
7	0.154V	0.196V	0.4595V
8	0.1693V	0.02702V	-0.2446V
9	0.01081V	0.01764V	-0.1369V
10	0.007V	0.014V	0.1367V
11	-0.267V	-0.25V	-0.281V
12	0.294V	0.361V	0.4881V
13	-0.016V	-0.07V	-0.2561V
14	0.167V	0.1738V	0.1434V
15	0.051V	0.063V	0.3237V
16	-0.226V	-0.239V	-0.4305V
17	-0.0669V	-0.583V	-0.29018V
18	0.0056V	0.0046V	-0.2474V

Table 5.2: Output voltages *three-wires* layout

Prediction error: definition and causes

Prediction errors arise the need for a deep understanding of such events' nature and possible causes. First, it is important to define what is an error in the neuromorphic computation under analysis. From the molecular point of view, the results are all correct since the final configuration present in the central cell is the one that provides the highest stability to the whole structure. This consideration implies that the simplicity of the adopted model limits our ability to track the correct output voltage value. There are two possible causes of prediction errors related to the model defined. The first one consists of not considering the shift of the bis-ferrocene transcharacteristics. Consider for instance simulation number nine in table 5.2. The predicted voltage is lower than 0.0395V, which is the value for which the charge distribution among the bis-ferrocene dots is null. Therefore, charges move toward the second dot of the first molecule, while the expectations point to the opposite case. This situation is verified every time the combination of the voltages gives rise to a final value lower than the VACT right shift. A possible solution could be to include in the model a shift of the weighted sum toward the left to ideally center the transcharacteristic around 0V. The corresponding mathematical equation is written in 5.1.

$$Vout = (w_{m1} \cdot V_{in1} + w_{m2} \cdot V_{in2} + w_{m3} \cdot V_{in3}) - V_{shift}^{bisfe} \tag{5.1}$$

A second reason can be identified in not taking into account the diagonal coupling contributions on the final interface polarization. In order to solve this problem, a superposition effect analysis is needed for each interface cell. The polarization on each molecule results from the combination of the input voltage value and the induced voltage through diagonal coupling. The final polarization on each interface can be evaluated as

$$P_{ind+dir} = f(V_{in}^{driver} + V_{diagonal}) \tag{5.2}$$

were *ind* stands for induced and *dir* for direct. Then, being the voltage at the output of each molecule a function of the polarization on the corresponding cell, the resulting voltage in input to the central cell is going to be

$$V_{in_x} = f(P_{ind+dir}) \tag{5.3}$$

The final proposed model is reported in the equation here below:

$$V_{out} = ((w_{m1} \cdot V_{in1}(P_{ind+dir}^{interface1}) + w_{m2} \cdot V_{in2}(P_{ind+dir}^{interface2}) + w_{m3} \cdot V_{in3}(P_{ind+dir}^{interface3})) - V_{shift}^{bisfe} \tag{5.4}$$

Concerning this thesis objective, this analysis is enough from the point of view of exact modeling. The proposed complex model can be a starting point for further studies in future research. In the rest of the thesis, we will deal with prediction errors trying to focus on how to avoid them from a circuit implementation point of view.

5.6 Final considerations for interconnections

In the course of this chapter, has been analyzed the best solution to implement neurons connected with long input wires. Once connected to the *additional cell* neurons, these two will be used to build the final structure of the neural network.

The analog solution was not feasible with the elements involved, so a digital implementation seems to be more reliable from the point of view of computation and information transport. Saturators are placed at the beginning of all the wires to ensure the binary behavior, and the intermolecular distance is equal to 0.9nm. The choice of such value

ensures reasonable weights for all the interfaces.
For what concern the clock structure and waveforms, the present study proposes a solution with at least four regions, with signals having proper time shift one to the others. Major attention must be paid to the saturator molecules activation to avoid loss of information due to back-propagation.
Several simulations have confirmed the validity of the proposed circuit and the compliance of the results to those found in the preceding chapters.

Chapter 6

Connecting neurons together

This thesis final goal is to propose a neural network with the circuit blocks defined so far. In this chapter, the network is built step by step by adopting all the basic structures that have been described. As a remark, the graphical result is the one reported in figure 6.1

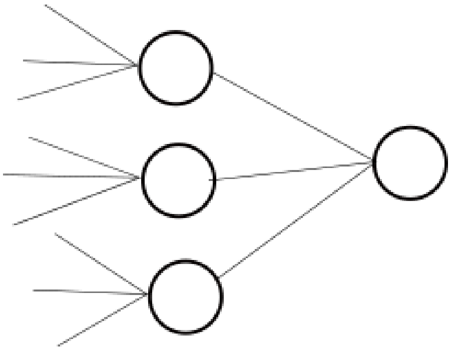


Fig. 6.1: Schematic of the neural network

The ANN is formed by one input layer made of three neurons and an output layer made of just one neuron. From a practical point of view, it is possible to adopt such a network to perform simple classification problems. The molecular implementation of the input layer will be made with *additional cell* neurons, whose output propagates along wires whose structure was discussed in Chapter 5.
The simulation arrangements reported in this chapter are taken from tables 5.2 and 5.1 to have a clear expectation and focus on the way the neurons have to be joined together from a structural and geometrical point of view.

6.1 First input neuron: connection with m3

The first elementary network consists of the simple connection within an *additional cell* input neuron with interface m3 of the output neuron. The schematic representation of the circuit is proposed in figure 6.2 which also shows the adopted clock structure. Consider that the schematic in 6.2 is highly ideal for what concerns the number of molecules present in each clock region. Concerning the clock signals, it is possible to rely on those presented in 5.27.

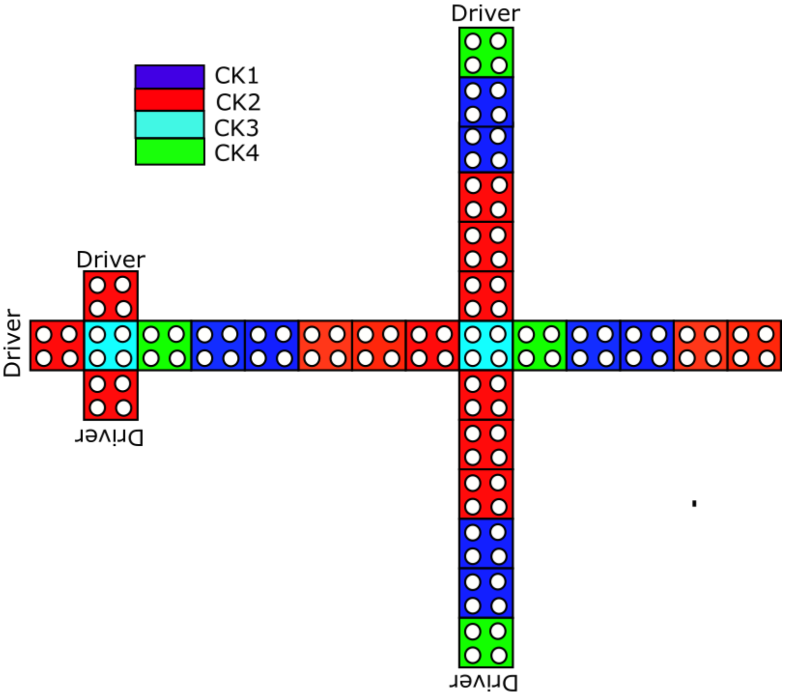
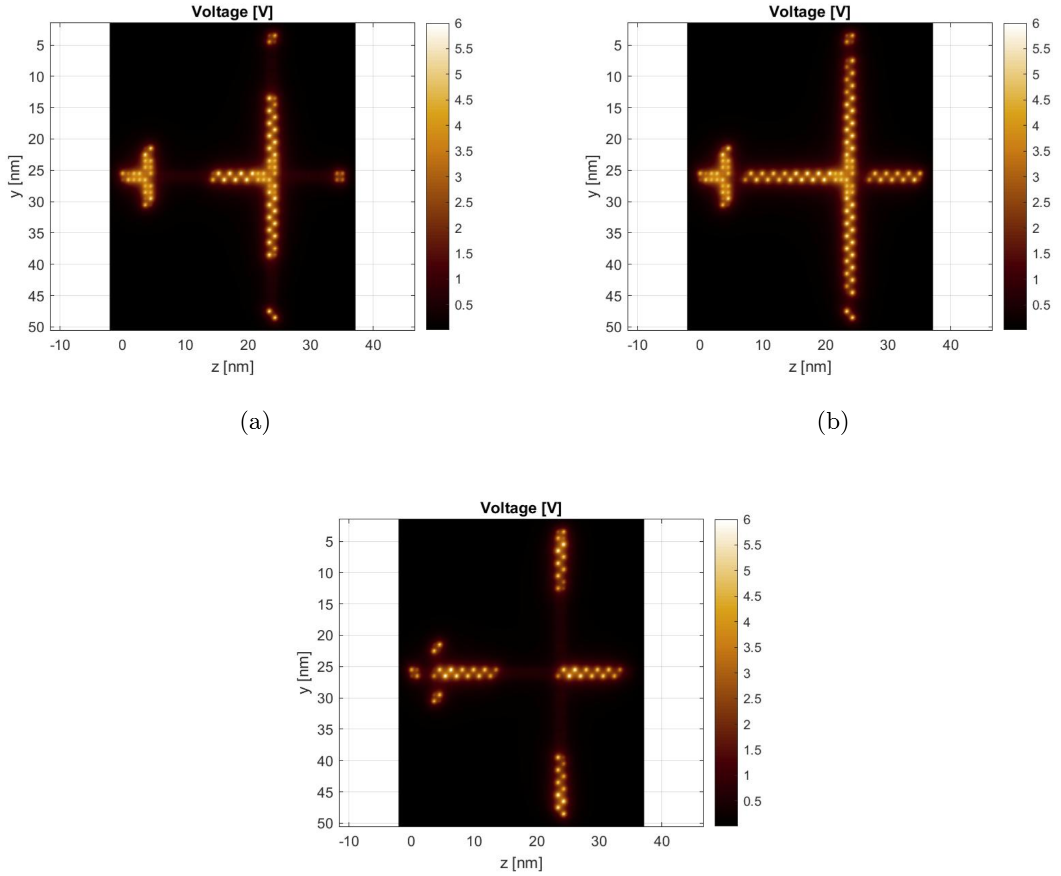


Fig. 6.2: Connections of two neurons. Clock regions distribution along the proposed circuit.

The wires are divided into two clock regions; as found in Chapter 5, this is a stable and safe operating solution. The interface molecules are placed in the second clock region, while the third clock signal activates the central cells. Then the saturators are located in the fourth clock region as the first cell of each wire. For what concern output layer's m1 and m2 interfaces, these are still driven by driver cells emulating the presence of neurons. Consider as a first example the input neuron to be configured as in simulation number two in 5.1, while the output neuron follows the exact configuration of simulation number twelve of the same table. A logic '1' should propagate to the output neuron from the input layer. The information coming from interface m2 is expected to be the prevailing one at the output. The results of this simulation are reported in figures 6.3.

From the snapshots, it is clear that the expectations are not respected, particularly concerning the output neuron, in which the inversion of the MV working behavior does not properly occur. It is therefore important to understand which are the causes. In figure 6.3b it is possible to see that the output wire, being placed partially in the first clock region, is already polarized before the arrival of the information. So the output neuron central cell will suffer from the back-propagation of the output wire, and if the polarization on the neuron is low, this will compromise the correct evaluation of the inputs, causing



(a)

(b)

(c)

Fig. 6.3: First connection simulation of two neurons together

information loss. To solve this problem have been provided two main modifications. The first one consists of changing the clock signal timing to activate the first region later in time. Until the fourth clock region molecules are not active and stable, CK1 stays low.
The second change implies increasing the number of cells placed in the fourth region. Therefore, there will be a higher separation between the output wire cells and the neuron body. The modified clock waveforms and region schematic are reported in figures 6.4 and 6.5.

The same simulation was performed again to prove the effectiveness of the proposed changes, with results reported in figures 6.6a and 6.6b. It is evident that the circuit works according to the expectations and the logic value in output to the circuit is a logic ‘1’. This is what can be expected looking at the results reported in tables 5.2 and 5.1.
This achievement is of particular importance since it is the first example of a correct connection within two neurons and the first step toward realizing the complete structure.

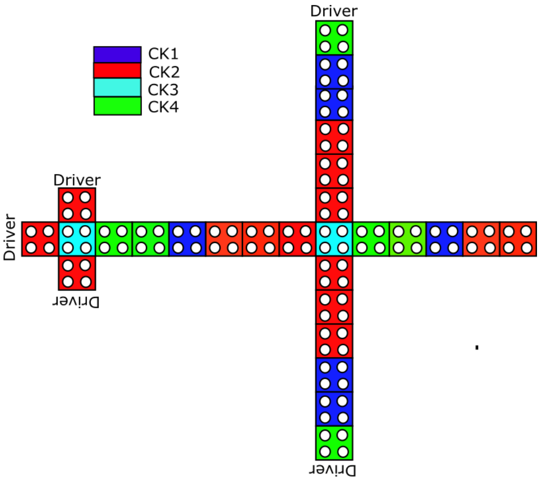


Fig. 6.4: Second clock region organization along the circuit

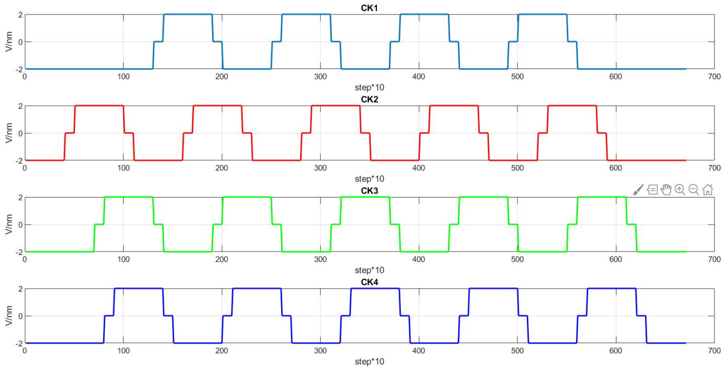
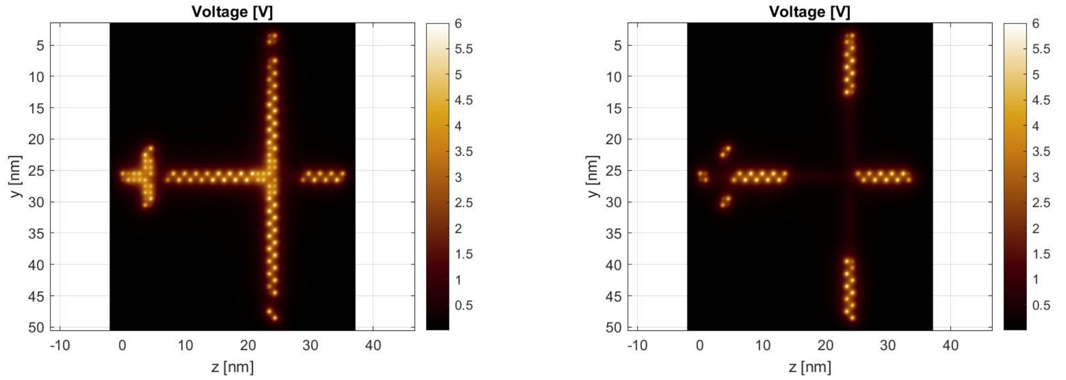


Fig. 6.5: Clock signals timing variation

6.1.1 Change the input values

Before moving on with the analysis of more complex architectures, it is worth spending some time analyzing the propagation of different pieces of information to verify the correct behavior of the circuit. The goal is to maintain the concepts of adiabatic propagation so that the inputs can change after resetting the neighboring cells. This way, it is possible to avoid propagation clashes and the consequent loss of the correct binary value. An example is reported in figures 6.7a-6.7d. On the input neuron, the configuration is presented in simulation number seven of table 5.2, while on the output layer, the interfaces are set as in simulation twelfth. The first neuron is subjected to the same voltage values as in the mentioned simulation for what concerns the inputs. On interfaces m1 and m2 of the output neuron are applied $V_{in_1} = V_{in_2} = 0.5V$. The final result is logic zero, which is congruent with the expectations. At the first neuron output, the firstly computed result moves along the output wire until it reaches the interface m3 of the second neuron. The



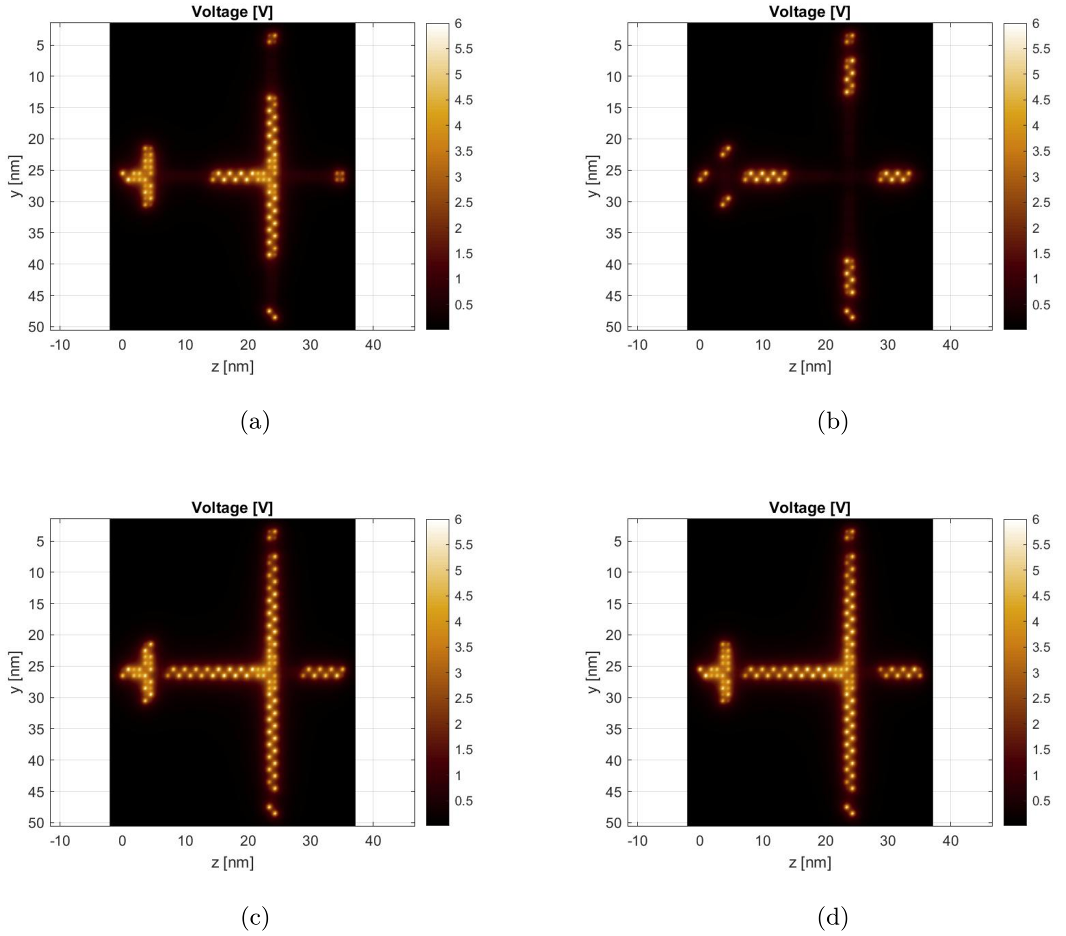
(a) Propagation of logic ‘0’ to the output neuron

(b) Propagation of logic ‘1’ computed in the output layer

Fig. 6.6: Propagation along the modified network

variation of the input values is visible in figure 6.7b. Notice that both the second and the third regions turn off simultaneously. Then, moving forward in time, these are activated again, and, after the second set of inputs has been correctly evaluated by the neuron, the new computed information can also propagate through the structure. At this point, an inversion behavior occurs for the new input values at the output neuron interfaces, as can be expected.

This straightforward example is another important step useful for building the final structure. The proposed structure correctly handles the presence of more than one set of inputs. As a last consideration, it is important to highlight the initial latency of the binary values before they propagate along the output wire of the circuit. The output is spurious and not of interest before the latency elapses.



(a)

(b)

(c)

(d)

Fig. 6.7: Input voltages variation during the propagation. The circuit behaves correctly

Voltage evaluation

This section proposes some hints about the voltage analysis in the two neuron structure under study. Even though, as already explained, the direction of this work is pointing toward digital applications, it is interesting to understand the analog behavior of the neurons and find possible differences with the circuits presented so far.
Many simulations have been performed, focusing the attention on the voltage levels on the first cell of the input neuron. The configurations adopted in these analysis are, once again, taken from tables 5.2 and 5.1. This way, it is possible to compare the results obtained on this circuit with those evaluated for the additional cell neurons. For what concerns the output layer, the voltage levels are comparable with the ones found using the *three-wires* neuron.
First, look at tables 6.1 for some examples.

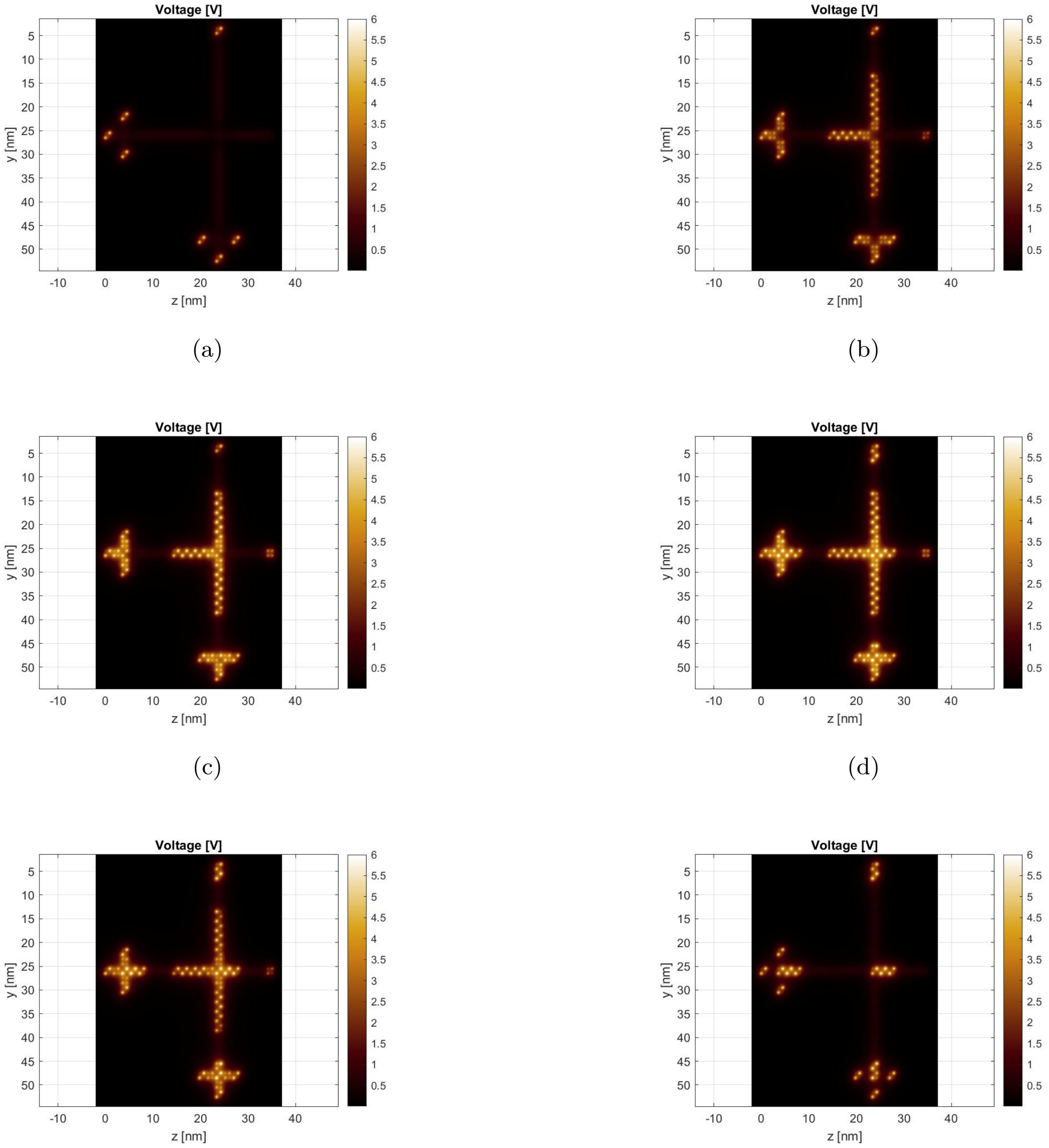
Output voltage results: input neuron on m3 interface			
Input layer	**Output layer**	**Expected V_{out}, input layer**	**Obtained V_{out}, input layer**
Simulation 3	Simulation 14	-0.0295V	-0.27316V
Simulation 5	Simulation 15	0.376V	0.49025V
Simulation 6	Simulation 15	-0.44979V	-0.5911V
Simulation 14	Simulation 15	-0.04342V	-0.25142V
Simulation 18	Simulation 5	-0.07495V	-0.2774V
Simulation 17	Simulation 4	-0.08118V	-0.42531V
Simulation 1	Simulation 5	0.11364V	0.32852V

Table 6.1: Output voltages on the central cell of the input neuron

From the values reported in the table above, it is clear that all the results are correct from a digital point of view. The circuit works as can be expected for all the proposed setups. The main problem is related to the difference between the expected V_{out} and the ones computed with this circuit.
The reason for that is the high number of molecules in the architecture under study. The number of electric field lines influencing the charges in each molecule is higher, thus producing a general increase in the charge distribution among the dots. According to FCN theory, it increases the voltages at the output of each cell.
At this point, the quality and correct ongoing of the circuit has been successfully demonstrated with two neurons connected.

6.2 Second input neuron: connection with interface m2

This section analyzes how a third neuron can be placed into the structure and connected to the output layer. The final circuit derived from this study consists of two *additional cell* neurons connected to the output layer. The third input of the output neuron is simply a driver whose logic value propagates through a molecular wire. Notice that this analysis can also be considered valid for what concerns interface m1 of the output neuron, being the structure symmetrical.
First thing is to identify the best way to place the neuron on the bottom wire connected to the output. A first assumption consisted in rotating the neuron layout by 90°. If this solution works correctly, no further modifications must be applied to the circuit structure. As a first case, consider the input voltages equal to 0.5V and all the interface molecules characterized by $\alpha = 1V$. The results are reported in figure 6.8.



(a) (b) (c) (d) (e) (f)

Fig. 6.8: Propagation through the three neurons network. Bottom neuron rotated by 90°

It is possible to state, at a first look, that the propagation is acceptable and the final computation works correctly. However, it is mandatory to highlight an essential thing in figure 6.8d. At the beginning of propagation on the vertical wire, charges move in the two upper dots of the cell to minimize Coulomb repulsion. It is an apparent effect of crosstalk and introduces uncertainty in the information propagation. Even if, in the end, the correct logic value is recovered and transmitted, it is better to avoid this kind of situation. So, the final structure should have a safe and easy-analyzable behavior. The crosstalk effect is even more evident if higher α molecules are involved. A highlight on the presence of crosstalk for such a neuron, having at the bottom an interface $\alpha = 2V$, is reported in figure 6.9.

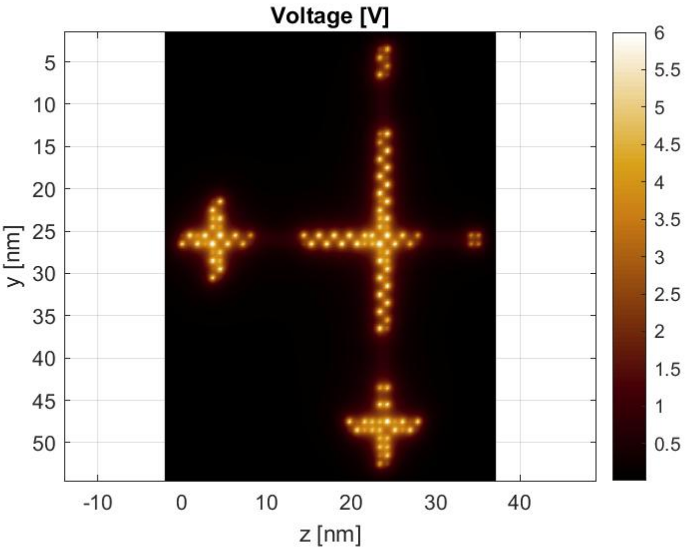


Fig. 6.9: Vertical neuron. Bottom interface: $\alpha = 2V$, lateral interfaces: $\alpha = 1V$

It is, therefore, evident that this proposed layout is not entirely correct. Indeed, even though all the performed simulations are correct from a binary point of view, it is better to avoid the presence of crosstalk as much as possible.

For this reason, the circuit layout has been modified both in circuit and clock characteristics. In figure 6.10, it is possible to appreciate the new circuit layout that will be used from now on. The propagation moves from left to right, compliant with usual applications of molecular FCN.

In the region of the bottom wire where there is a change of direction from horizontal to vertical, it is possible to observe a typical solution adopted to avoid loss of information. A fifth clock signal has been added to provide transport as stable as possible, and it has to drive the cells in the bottom corner. Notice that the first cell of the vertical wire can also be included in this region, thus providing further reliability to the circuit. As already happened for the fourth clock signal, adding the fifth signal was not straightforward. The main problem is transporting the information from clock region number one (yellow cells) and moving it to the first (light green cells). Nothing changes for what concerns the other components: the interfaces are controlled by CK2, while CK4 controls the saturator molecules at the beginning of each wire.

In conclusion, with the clock signals and timing reported in figure 6.11, it is possible

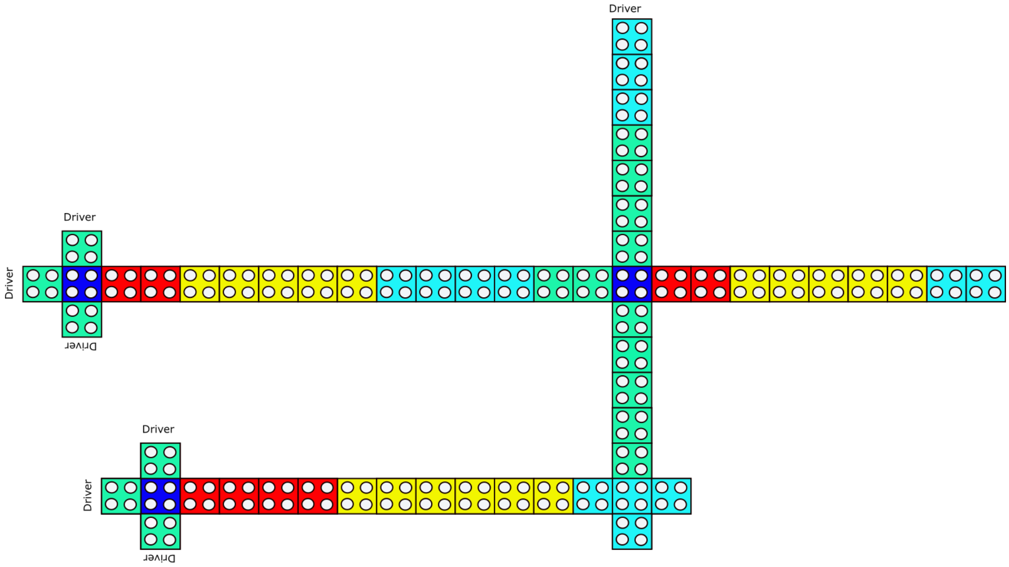


Fig. 6.10: Three neurons network schematic

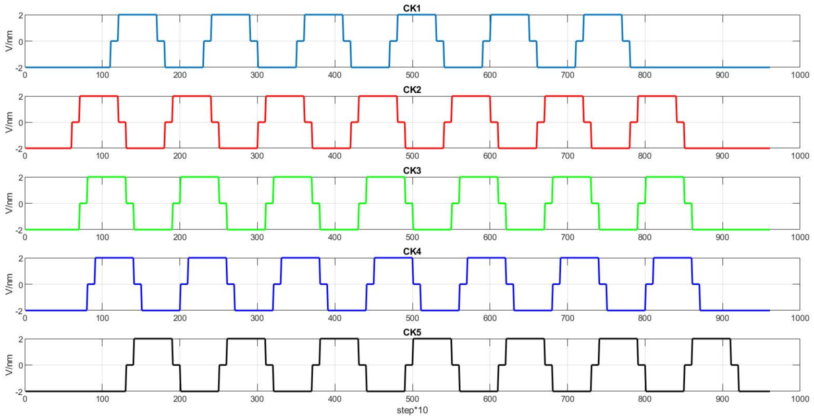


Fig. 6.11: Final working clock profile

to propagate every information through the circuit correctly. This layout will also be adopted for the final addition of the third input neuron on interface m1.
For the sake of simplicity, from now on, the neurons in the structure will be referred to in this way: the neuron connected to interface m3 of the output layer is named *Neuron A*, the other two on m2 and m1 are going to be, respectively, *Neuron B* and *Neuron C*.
Consider now the situation reported below:

- **Neuron A**:
  - m3: $\alpha = 1V$, $V_{in_3} = -0.5V$

  - m2: $\alpha = 1V$, $V_{in_2} = -0.5V$
  - m1: $\alpha = 1V$, $V_{in_1} = -0.5V$

- **Neuron B**:
  - m3: $\alpha = 1V$, $V_{in_3} = -0.5V$
  - m2: $\alpha = 1V$, $V_{in_2} = -0.5V$
  - m1: $\alpha = 1.5V$, $V_{in_1} = -0.5V$

- **Output neuron**: all interfaces $\alpha = 1V$

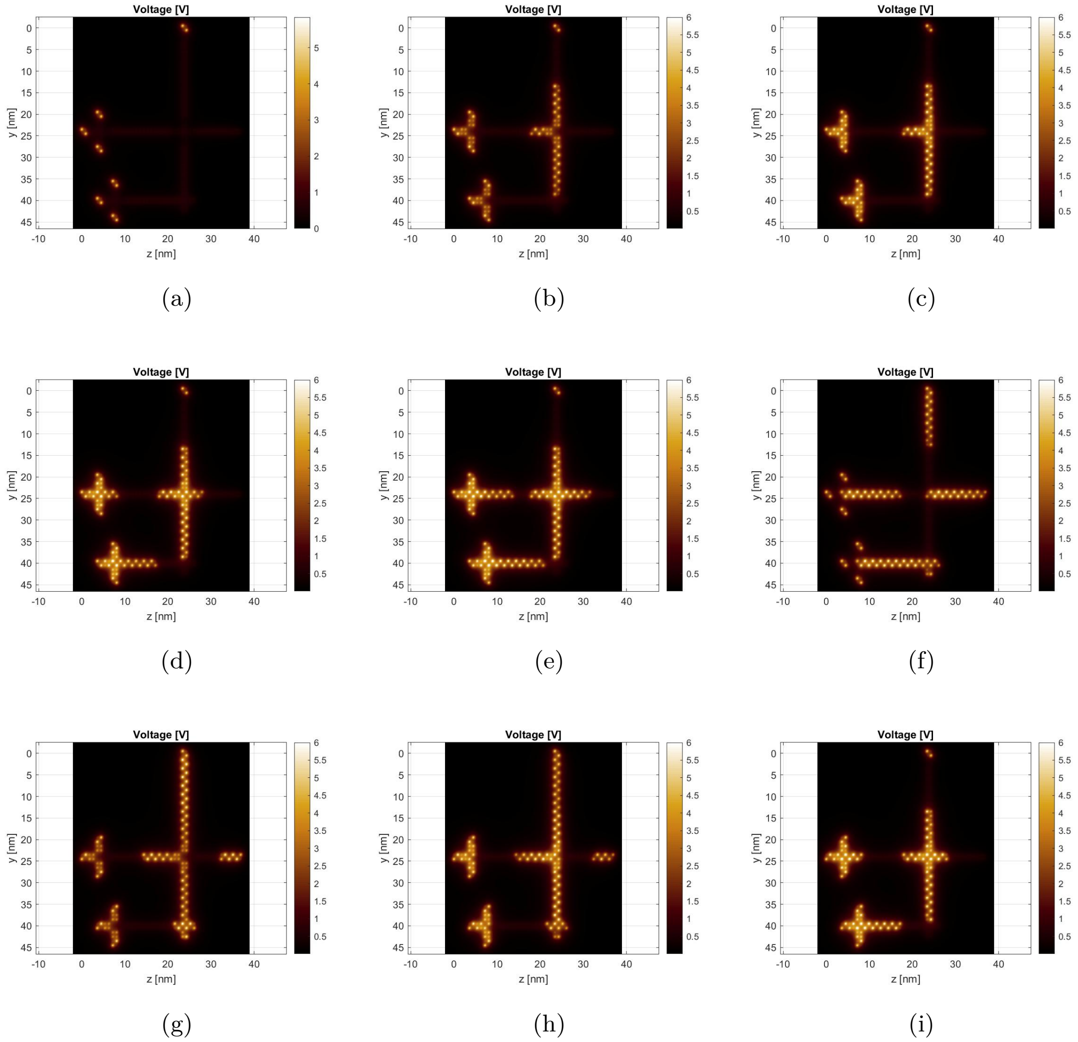
It is a straightforward configuration concerning the interfaces and the input voltages. However, it will be helpful to understand the information propagation behavior and verify the neurons' independence. This last consideration is crucial. Indeed, the layout must guarantee that the neurons on the input layer can compute the corresponding output values without considering external influences, i.e., electric field lines coming from the rest of the circuit. This is obvious regarding the output neuron, while the design must involve a certain amount of space between Neuron A and Neuron B active regions. The first results obtained with this circuit in the configuration explained before are reported in figures 6.12a-6.12i.

The computation is correct in all the neurons, and it is possible to demonstrate that the voltage levels in output to the central cells of A and B are equal to those found in the two neurons circuit, thus eliminating the risk of spurious electric field couplings. After the neurons computation, the information propagates through the circuit. The correct organization of the clock regions avoids clashes and interference problems. The change in the direction is correctly performed in the bottom wire, thanks to the proper timing of CK5. Finally, the output layer accurately evaluates the incoming binary values, and the final result can be transported along the output wire.

Once again, have been performed an analysis related to the voltage levels according to the basic formula 3.2, the results are compared with the previously obtained ones to highlight possible critical setups of the interfaces. The driver connected to interface m1 is fixed at 0.5V.

The output neuron has all the interfaces with $\alpha = 1V$. Some numerical results taken from the input neurons are written in tables 6.2 and 6.3. The red-colored cells indicate an error in the result prediction, which also occurs in these simulations.

The advantage of having an output layer in the network is the possibility of avoiding the effects of prediction errors in one of the three input neurons. Consider, for instance, the fifth simulation in table 6.3; in Neuron B, the expected binary value was a logic '0'. Instead, a logic '1' is computed and transported through the network. Ideally, the input at the interfaces of the output neuron should be '000', respectively, for m1, m3, and m2. It would give a final output value corresponding to logic '0'. With the actual values computed by the neurons, the input set corresponds to '001', which provides logic '0' as well at the output due to the presence of $\alpha = 1V$ molecules on all the output neuron sides. This example is a clear case in which possible mismatches within simulation and predictions can be solved by proper configuration of the output layer and their possible



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Fig. 6.12: Working solution to propagate informations along a circuit with three neurons

effects nullified. The key point is the identification of possible critical situations through the independent analysis of neurons and the proper configuration of the weights at the output interfaces.

On the other hand, in some cases, as the third simulation in table 6.3, the prediction error would lead to designing a network computing wrong results. In that case, always considering the configuration presented before, the input set to the output layer is ‘101’, giving a logic ‘1’ while the expectations were toward a logic ‘0’.

Therefore, the main phase in building a neural network in this way consists of identifying the expected behavior of the output and, through single neuron analysis, configuring all the subcircuits according to the needs.

Neuron A			
m3	**m2**	**m1**	V_{out}
-0.5V	0.8V	-0.5V	0.09364V
0.5V	0.5V	0.5V	0.36695V
0.8V	-0.5V	-0.4V	-0.29863V
0.7V	-0.5V	0.3V	-0.05315V
0.5V	0.7V	-0.5V	0.08351V
-0.2V	-0.5V	0.5V	-0.00506V
0.5V	0.8V	0.5V	0.47324V
$\alpha = 2V$	$\alpha = 1V$	$\alpha = 1V$	

(a) First simulation: Neuron A

Neuron B			
m3	**m2**	**m1**	V_{out}
-0.7V	0.5V	0.8V	0.08742V
0.5V	-0.4V	0.5V	0.0519V
0.4V	-0.7V	-0.8V	-0.12699V
0.8V	-0.8V	0.2V	0.00162V
-0.5V	0.2V	-0.5V	-0.0708V
0.8V	0.6V	-0.3V	0.07867V
0.5V	-0.8V	0.5V	0.0141V
$\alpha = 1.5V$	$\alpha = 2.5V$	$\alpha = 2V$	

(b) First simulation: Neuron B

Table 6.2: Output voltages on the central cells. First simulation

Neuron A			
m3	**m2**	**m1**	V_{out}
-0.5V	-0.5V	-0.5V	-0.4867
0.5V	0.5V	0.5V	0.4867V
0.5V	-0.5V	-0.5V	-0.0307V
0.5V	-0.5V	0.5V	0.3236V
0.5V	0.5V	-0.5V	0.1324V
-0.5V	-0.5V	0.5V	-0.1324V
-0.5V	-0.5V	0.5V	0.3236V
$\alpha = 1V$	$\alpha = 1.5V$	$\alpha = 1V$	

(a) Second simulation: Neuron A

Neuron B			
m3	**m2**	**m1**	V_{out}
-0.5V	-0.5V	-0.5V	-0.49225V
0.5V	0.5V	0.5V	0.4524V
0.5V	-0.5V	-0.5V	0.0036V
0.5V	-0.5V	0.5V	0.3579V
0.5V	0.5V	-0.5V	0.0981V
-0.5V	-0.5V	0.5V	-0.0981V
-0.5V	-0.5V	0.5V	0.3579V
$\alpha = 1V$	$\alpha = 2.5V$	$\alpha = 1V$	

(b) Second simulation: Neuron B

Table 6.3: Output voltages on the central cells. Second simulation

6.3 Third input neuron: connection with m1

It is necessary to add the third input neuron to complete the layout of the neural network: *Neuron C* connected to the m1 interface of the output layer.
The operation is relatively straightforward since it is a replica of the circuit introduced on interface m2. Being these two symmetrical from a circuit point of view, it is enough to report the same cell organization adopted before. The final circuit schematic of the two-layer neural network is reported in figure 6.13. Notice that, as anticipated, at both the corners of the input wires, a further cell is present to improve the propagation stability. This is the final structure of the neural network and one of the basic building blocks present in more complex circuits proposed in the following chapter.

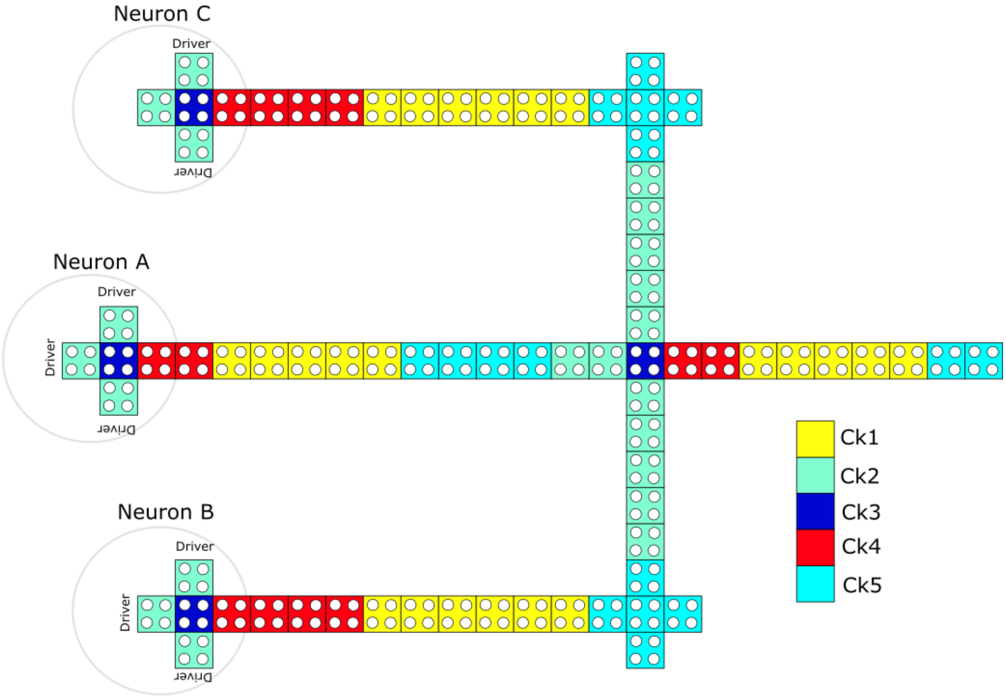


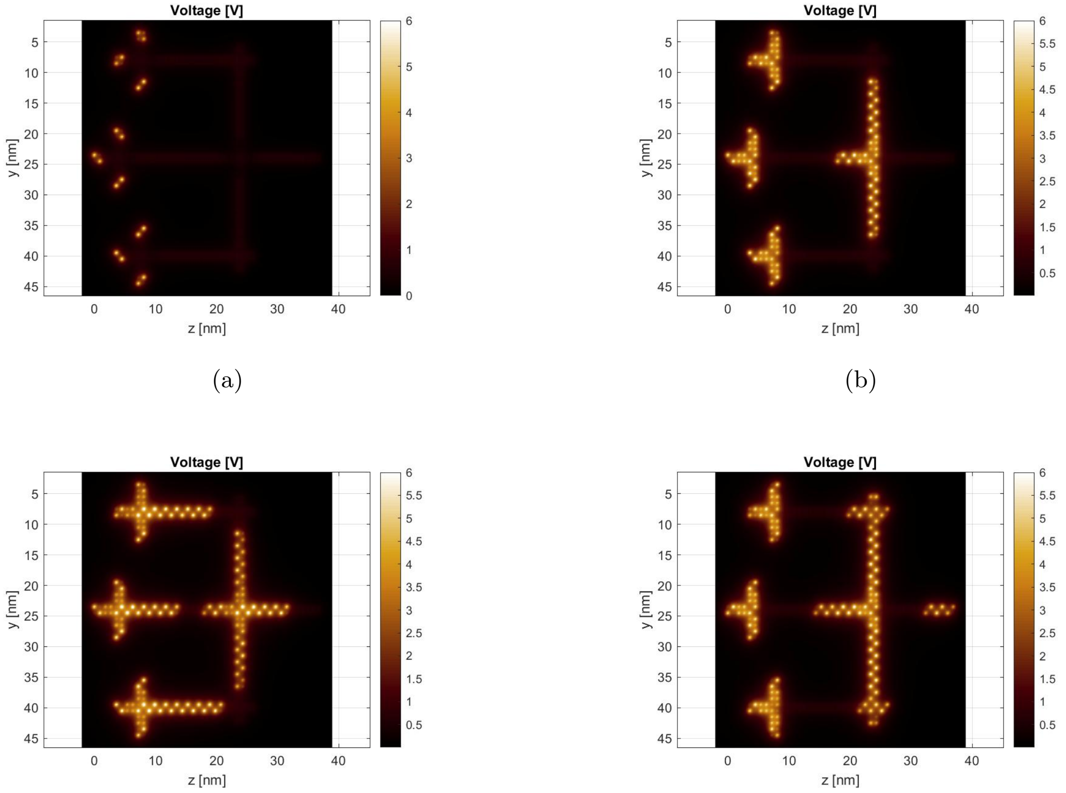
Fig. 6.13: Four neurons network schematic

From a propagation point of view, nothing has to be added since, as said, the cells added on interface m1 behave in the same way analyzed for the bottom connection.
It is interesting to verify simulation results to prove the correctness of the proposed solution. A first one is reported in figures 6.14a-6.14d. The configuration of the different blocks in the circuit is the following:

- **Neuron A:**
  - m3: $\alpha = 2.5V$, $V_{in} = -0.5V$
  - m2: $\alpha = 1V$, $V_{in} = 0.8V$
  - m1: $\alpha = 2V$, $V_{in} = -0.5V$

- **Neuron B:**
  - m3: $\alpha = 2.5V$, $V_{in} = -0.7V$
  - m2: $\alpha = 1.5V$, $V_{in} = 0.5V$
  - m1: $\alpha = 1.5V$, $V_{in} = 0.8V$
- **Neuron C:**
  - m3: $\alpha = 1V$, $V_{in} = 0.4V$
  - m2: $\alpha = 2.5V$, $V_{in} = 0.6V$
  - m1: $\alpha = 2.5V$, $V_{in} = -0.3V$
- **Output Neuron:**
  - m3: $\alpha = 1V$
  - m2: $\alpha = 1V$
  - m1: $\alpha = 1V$

Focusing the attention on the propagation itself, it is possible to state that everything works as expected. Moreover, considering the coefficients reported in 4.4, the logic value obtained at the network's output is in line with the expectations.
As a second topic, it is interesting to go more into the detail about the error suppression capabilities of the output layer. To do such an analysis, consider the input sets in tables 6.4 - 6.7, in which are reported the interface configuration and input sets respectively for Neurons A, C, and B. The setup of the input layer is exactly the one presented previously, while the output neuron has $\alpha = 1.5V$ on m3, $\alpha = 1V$ on m2, and $\alpha = 3V$ on m1. This last configuration represents the correction applied to the structure to minimize the error rate. In the tables, the rows highlighted in red represent an error in evaluating the output values in the input layer. However, it is possible to appreciate that the errors can be suppressed with the adopted interfaces on the output layer. This is a strong demonstration of the possibility of reducing the error rate of the network by a proper combination of the available weights. Unfortunately, the correction is not successful for all the input sets, meaning that probably further modifications would be needed. Indeed, in table 6.7, the last row, colored in red, indicates the presence of a mismatch between the prediction and the obtained values. In figures 6.15a-6.15b are reported, respectively, the first and the last propagation time instants for the last input set reported in the tables below.



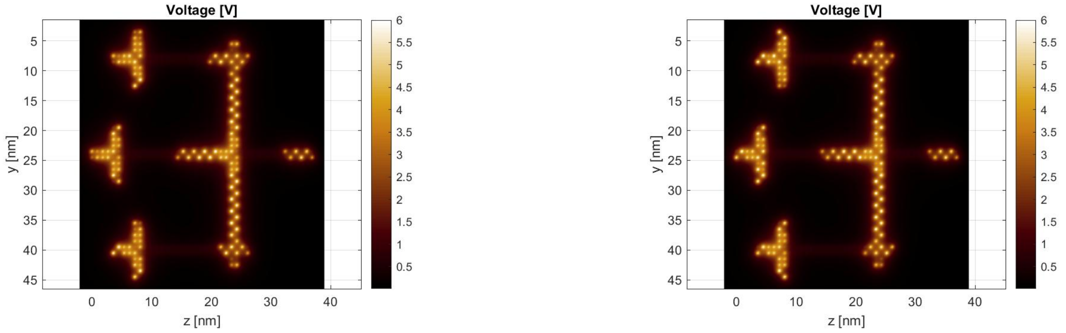
(a)

(b)

(c)

(d)

Fig. 6.14: Complete neural network: three neurons on the input layer and one neuron on the output layer



(a)

(b)

Fig. 6.15: Complete neural network: configuration present in the last row of table 6.4. First and last steps in the propagation

Neuron A			
m3: $\alpha = 2.5V$	m2: $\alpha = 1V$	m1: $\alpha = 2V$	V_{out}
-0.5V	0.8V	-0.5V	0.18324V
0.5V	0.5V	0.5V	0.27735V
0.8V	-0.5V	-0.4V	-0.14943V
0.7V	-0.5V	0.3V	-0.08107V
0.5V	0.7V	-0.5V	0.23771V
-0.2V	-0.5V	0.5V	-0.13988V

Table 6.4: Neuron A

Neuron C			
m3: $\alpha = 1V$	m2: $\alpha = 2.5V$	m1: $\alpha = 2.5V$	V_{out}
0.4V	0.6V	-0.3V	0.2345V
0.3V	0.4V	0.5V	0.18225V
-0.7V	-0.3V	-0.4V	-0.3537V
0.1V	-0.1V	0.2V	0.03921V
-0.4V	0.5V	0.6V	-0.12957V
-0.3V	0.7V	0.2V	-0.06759V

Table 6.6: Neuron C

Neuron B			
m3: $\alpha = 2.5V$	m2: $\alpha = 1.5V$	m1: $\alpha = 1.5V$	V_{out}
-0.7V	0.5V	0.8V	0.14046V
0.5V	-0.4V	0.5V	0.05586V
0.4V	-0.7V	-0.8V	-0.2001V
0.8V	-0.8V	0.2V	-0.0281V
0.8V	-0.8V	0.2V	-0.08848V
0.8V	0.6V	-0.3V	0.12409V

Table 6.5: Neuron B

Final output logic values		
Expected	Obtained without correction	Obtained with correction
0	0	0
0	0	0
1	1	1
1	1	1
1	0	1
1	0	0

Table 6.7: Output logic values: the error rate reduces thanks to the proper choice of the output neuron interfaces

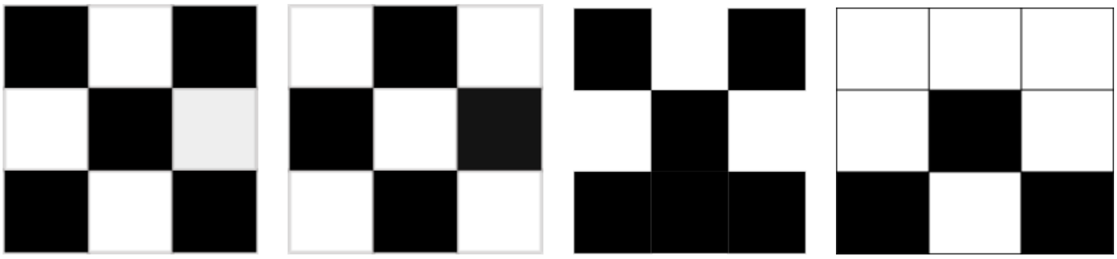
Chapter 7

Pattern recognition using molecular FCN neural networks

A whole set of grounding rules and considerations have been provided in the previous chapters. According to those, it is possible to build a functional neural network based on molecular Field-Coupled Nanocomputing.
Until now, the analyses carried out on the circuits focused the attention on the voltages and possible prediction error evaluations. It is then essential to move forward from the analog domain related to the topic and put some effort into realizing digital networks able to perform real-field scenario tasks. A new network will be defined starting from the considerations performed in the last part of this thesis. The circuit will perform simple pattern recognition tasks, and the results will be compared with those obtained through a software-trained feed-forward neural network.

7.1 Pattern recognition

First, it is essential to define the task that the neural network has to accomplish. The choice was to build a circuit that correctly classifies four different patterns arranged in 3x3 matrix arrays. It is a common application field for feed-forward neural networks [58], as those feasible with molecular technology so far. Even though such networks are adopted in complex real-world situations, such as medical and financial, we are limiting, as said, in recognizing just a limited number of paths and, according to that, provide proper logic values at the circuit output. However, the results that will be shown offer a first and unambiguous indication of the functionalities of the proposed solution.
The four patterns to be classified are reported in figures 7.1a-7.1d.



(a) Pattern A (b) Pattern B (c) Pattern C (d) Pattern D

Fig. 7.1: 3x3 binary images

These simple matrices will represent the inputs to the neural network, which has to provide the correct outputs depending on the pattern under analysis. In particular, two bits are needed to recognize four different input combinations in a logic domain. Therefore, two bit parallelism must be guaranteed at the circuit output.
In addition, it is of primary importance to define criteria to apply the correct voltage levels according to the encoded pixel. The choice consists of associating to the black colored pixels a logic '1' and a '0' to the white ones. From the molecular point of view, the previous chapters explained the association between logical '1' and '0' to negative and positive input voltages, respectively. In particular, for this analysis, voltages show absolute values equal to 0.5V; in this way, it was possible to provide flexibility to the neurons working behavior. The just explained procedure can be considered compliant with the nowadays available solutions for image recognition applications based on deep learning [59]. Given the high number of colors in an input image, analog values are associated with each pixel. For this thesis purposes, the digital implementation developed in this section is enough and provides a first step toward implementing molecular FCN neural networks.
Concerning the outputs, the desired values are those presented in table 7.1. Notice that patterns C and D can be though as modifications of pattern A. Therefore, they must be classified in a different way. Some more analysis related to variations of input patterns will be provided in the following.

Pattern	Output1	Output2
A	1	0
B	0	1
C	1	1
D	0	0

Table 7.1: Expected truth table of the final circuit

7.2 Software trained neural networks

As anticipated at the beginning of this chapter, a comparison will be performed between the neural network outcomes and those obtained through software simulations. The scope

is to evaluate the possibilities regarding the number of neurons involved in the two solutions to reach the same results.
The software network is feed-forward and fully connected; this choice allows for maintaining the programming complexity as low as possible. The goal network is schematically reported in figure 7.2.

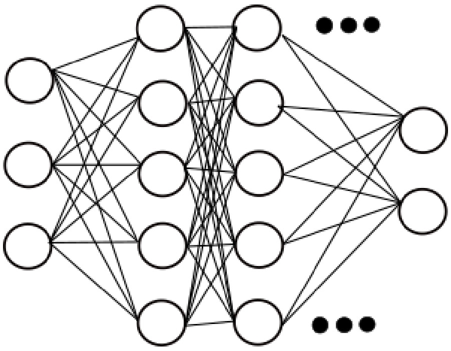


Fig. 7.2: Feed-forward fully connected neural network

From the schematic, it is clear that there are an input and output layer and a certain number of hidden layers, whose amount has to be defined according to the training phase considerations. Usually, these networks topologies are well suited for classification purposes [59].
So, the first step in the development procedure was to decide the kind of algorithm needed to train the neural network. Considering that the amount of possible solutions is huge, the choice was to start with the more classical and easy to be implemented: the back-propagation algorithm [60].
The algorithm has been implemented using Matlab. Even though Mathworks offers a working toolbox to design and simulate neural networks [61], an ad-hoc code has been preferred. In this way, it was possible to study and go into the detail of the design procedure of the network itself.
The program is reported in the boxes in the following pages and consists of three main parts:

- Training script training the network to recognize a pre-defined number of patterns
- Deep Learning script implementing the back-propagation and delta rule algorithms
- Test script to verify the network functionalities

The training and deep learning codes are reported below in the two boxes. First, the input patterns are defined, while the *correct output* matrix associate each input to the proper output bits. The weights for each neuron in each layer are randomly defined according to the code line.

```
w_x = 2*rand(n, y)
```

Where **n** is the number of neurons and **y** is the number of inputs to each neuron. The code also allows uploading already evaluated weights through an *if* statement.
The main core of the training phase is the *for loop* at the end of the script.
It is important to notice that to achieve high accuracy in the results, there are two main parameters: the number of epochs and the number of neurons per layer. The first is simply the number of times the network is trained, improving the weight evaluation. For each iteration, the **deep learning** function is recalled. It implements the back-propagation rule and can be analyzed in the second box. The number of patterns to be recognized represents the amount of iteration of a *for loop* updating the weights values in each epoch. The output of each layer is computed according to the inputs, the current weights, and the activation function. In this case, the activation function is the same for all the neurons and is of the REctified Linear Unit (ReLU) type. Figure 7.3 reports its graphical characteristics.

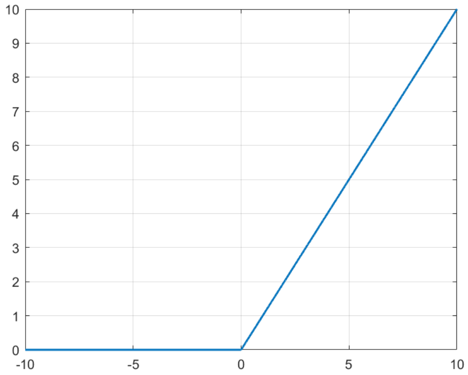


Fig. 7.3: ReLU activation function

The error is the difference between the correct output and the obtained one.
After that, the *delta rule* algorithm can start and leads to the weight adjustments through the back-propagation of the errors from the output to the input layer. Notice that the weight coefficients correction rate is governed by **alpha** parameter; the smaller it is, the longer and more precise will be the training procedure. The weight adjustment formula is the following:

$$W_{x,t+1} = W_{x,t} + \alpha \cdot \delta \cdot outputLayer_{x-1,t} \tag{7.1}$$

Where $W_{x,t}$ is the weight vector of layer x at time t+1, α is the updating coefficient, δ represents the propagated error and $outputLayer_{x-1,t}$ is the vector representing the outcome of hidden layer number x-1 at time t. The formula is quite easy to understand and works for each layer in the neural network.
As the last point, it is worth providing some words related to the testing procedure. The code is reported in the third box on the following pages.

The script loads the *.mat* file produced as an output in the previous stage; this contains all the weights for each layer organized in matrix structures. Then, the rest of the code allows for selecting one among the four images and provides it at the neural network inputs. The pieces of information propagate along with the structure according to the weighted inputs formula and the chosen activation function.

Besides the awareness of the simplicity of the proposed code, it was extremely helpful in understanding the general working behavior of such a neural network. Moreover, it was possible to have a rough idea of the needs regarding the number of neurons and input handling for the realization of the molecular FCN circuit.

```
Training Network

input_image(:,:,1) = [0 1 0; 1 0 1; 0 1 0;];
input_image(:,:,2) = [1 0 1; 0 1 0; 1 0 1;];
input_image(:,:,3) = [1 0 1; 0 1 0; 1 1 1;];
input_image(:,:,4) = [0 0 0; 0 1 0; 1 0 1;];
correct_output= [0 1; 1 0; 1 1; 0 0;];
new_set = 1;
if new_set == 1
    n = 20;
    w1 = 2*rand(n,9)-1; w2 = 2*rand(n,n)-1;
    w3 = 2*rand(n,n)-1; w3_bis = 2*rand(n,n)-1;
    w4 = 2*rand(n,n)-1; w5 = 2*rand(n,n)-1;
    w6 = 2*rand(2,n)-1;
else
    w1 = dlmread("w1.txt"); w2 = dlmread("w2.txt");
    w3 = dlmread("w3.txt"); w3_bis = dlmread("w3_bis.txt");
    w4 = dlmread("w4.txt"); w5 = dlmread("w5.txt");
    w6 = dlmread("w6.txt");
end
for epoch = 1:800000
    [w1, w2, w3, w3_bis,  w4, w5, w6] =
    deeplearning(w1, w2, w3, w3_bis, w4,
                w5, w6, input_image, correct_output);
end

save('DeepNeuralNetwork.mat');
```

```
Back-propagation and Delta rule

function[w1, w2, w3, w3_bis, w4,w5,w6] =
          deeplearning(w1, w2, w3, w3_bis,
               w4,w5,w6, input_image, correct_output)
alpha = 0.0001; %% control learning rate
N = 4; %four patterns digits we want to recognized
for k = 1:N
```

```
    reshaped_input_image = reshape(input_image(:,:,k), 9,1 );
    input_of_hidden_layer1 = w1*reshaped_input_image;
    output_of_hidden_layer1 = ReLu(input_of_hidden_layer1);

   input_of_hidden_layer2 = w2*output_of_hidden_layer1;
     output_of_hidden_layer2 = ReLu(input_of_hidden_layer2);
%
    input_of_hidden_layer3 = w3*output_of_hidden_layer2;
     output_of_hidden_layer3 = ReLu(input_of_hidden_layer3);

     input_of_hidden_layer3_bis = w3_bis*output_of_hidden_layer3;
     output_of_hidden_layer3_bis = ReLu(input_of_hidden_layer3_bis);

    input_of_hidden_layer4 = w4*output_of_hidden_layer3_bis;
     output_of_hidden_layer4 = ReLu(input_of_hidden_layer4);

     input_of_hidden_layer5 = w5*output_of_hidden_layer4;
     output_of_hidden_layer5 = ReLu(input_of_hidden_layer5);

    input_of_output_node = w6*output_of_hidden_layer5;
    final_output = ReLu(input_of_output_node);

    correct_output_transpose = correct_output(k,:)'; %single column image
    error = correct_output_transpose - final_output;

    delta = error;   %% needed to apply delta rule to %%evaluate the weights
        %%of the hidden layers
    error_of_hidden_layer5 = w6'*delta;
    delta5 = (input_of_hidden_layer5>0).*error_of_hidden_layer5;

    error_of_hidden_layer4 = w5'*delta5;
    delta4 = (input_of_hidden_layer4>0).*error_of_hidden_layer4;

    error_of_hidden_layer3_bis = w4'*delta4;
    delta3_bis = (input_of_hidden_layer3_bis>0).*error_of_hidden_layer3_bis;

        error_of_hidden_layer3 = w3_bis'*delta3_bis;
    delta3 = (input_of_hidden_layer3>0).*error_of_hidden_layer3;

    error_of_hidden_layer2 = w3'*delta3;
    delta2 = (input_of_hidden_layer2>0).*error_of_hidden_layer2;

    error_of_hidden_layer1 = w2'*delta2;
    delta1 = (input_of_hidden_layer1>0).*error_of_hidden_layer1;

    adjustment_of_w6 = alpha*delta*output_of_hidden_layer5';
    adjustment_of_w5 = alpha*delta5*output_of_hidden_layer4';
    adjustment_of_w4 = alpha*delta4*output_of_hidden_layer3_bis';
    adjustment_of_w3_bis = alpha*delta3_bis*output_of_hidden_layer3';
    adjustment_of_w3 = alpha*delta3*output_of_hidden_layer2';
    adjustment_of_w2 = alpha*delta2*output_of_hidden_layer1';
    adjustment_of_w1 = alpha*delta1*reshaped_input_image';
    w6 = w6+adjustment_of_w6; w5 = w5+adjustment_of_w5;
    w4 = w4+adjustment_of_w4; w3_bis = w3_bis+adjustment_of_w3_bis;
    w3 = w3+adjustment_of_w3; w2 = w2+adjustment_of_w2;
    w1 = w1+adjustment_of_w1;
    end
end
```

```
Test Neural Network

load('DeepNeuralNetwork.mat');

input_image(:,:,1) = [0 1 0; 1 0 1; 0 1 0;];
input_image(:,:,2) = [1 0 1; 0 1 0; 1 0 1;];
input_image(:,:,3) = [1 0 1; 0 1 0; 1 1 1;];
input_image(:,:,4) = [0 0 0; 0 1 0; 1 0 1;];
 chosen_image = input_image(:,:,4);

 chosen_image = reshape(chosen_image, 9,1);
 input_of_hidden_layer1 = w1*chosen_image;
    output_of_hidden_layer1 = ReLu(input_of_hidden_layer1);

    input_of_hidden_layer2 = w2*output_of_hidden_layer1;
    output_of_hidden_layer2 = ReLu(input_of_hidden_layer2);
%
    input_of_hidden_layer3 = w3*output_of_hidden_layer2;
     output_of_hidden_layer3 = ReLu(input_of_hidden_layer3);

     input_of_hidden_layer3_bis = w3_bis*output_of_hidden_layer3;
     output_of_hidden_layer3_bis = ReLu(input_of_hidden_layer3_bis);

     input_of_hidden_layer4 = w4*output_of_hidden_layer3_bis;
     output_of_hidden_layer4 = ReLu(input_of_hidden_layer4);

     input_of_hidden_layer5 = w5*output_of_hidden_layer4;
     output_of_hidden_layer5 = ReLu(input_of_hidden_layer5);

    input_of_output_node = w6* output_of_hidden_layer5;
    final_output = ReLu(input_of_output_node);
```

7.2.1 Software results for the proposed task

Therefore, the software network was trained to solve the proposed pattern recognition problem. It is essential to highlight the conditions in which a correct working behavior, i.e., correct classification of the inputs, has been obtained. First, the number of information to the first layer of the network equals nine, i.e., the number of digits in which each pattern can be decomposed. Several trials have been performed to identify the correct balance between the number of epochs and the neurons in each layer, considering that just two neurons form the output layer. Furthermore, the minimum number of neurons in the hidden layers is equal to twenty. The neurons organize into one input and output layer and five hidden layers. The network has been trained for 800000 epochs to reach weight coefficients satisfying the needs.

7.3 Molecular FCN neural classifier

The implementation of a molecular FCN classifier is not straightforward at all. The main reason for that is the impossibility of organizing the molecular structure in the same way done in the software network. Indeed, the type of neurons with which it is possible to work with has three inputs, thus eliminating the possibility of working with fully connected circuits. Another problem arises considering full connection among neurons: the molecular wires would become high in number, extremely difficult to separate, and cross-wires would be needed. Unfortunately, cross-wire structures are still under preliminary analysis and could not be used for this task. Another difference to the software-trained network is the molecules activation function, which has a tansigmoid behavior [62].
As said in the previous section, the number of inputs to the structure equals nine. So, the first step consists of designing a molecular structure that takes nine different values as inputs of the circuit. For this reason, has been introduced the concept of *macro-neurons.* Figure 7.4 reports the circuit block schematic.

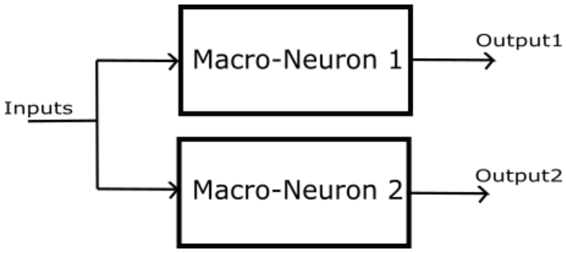


Fig. 7.4: Final circuit schematic block diagram

Each *macro-neuron* takes nine inputs and provide a different output, according to the truth table reported in 7.1. Given that each macro-neuron is an independent neural network providing a different output value with equal inputs, two different combinations of molecules must be found to compute the outputs properly.

7.3.1 Architecture analysis of the molecular circuit

In figure 7.5, it is possible to appreciate the circuit layout of the single macro-neuron, combined with how the inputs stimulate the structure.

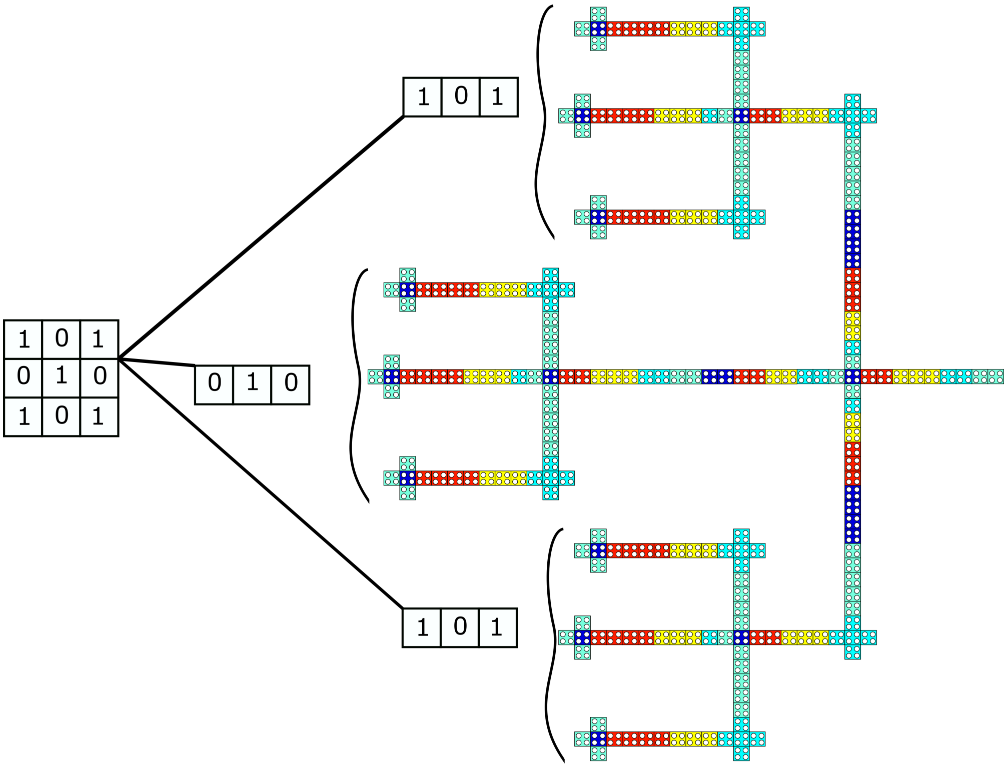


Fig. 7.5: Macro-neuron layout and inputs organization

The circuit can be divided into three equal parts showing the same layout defined at the end of the discussion in Chapter 6. The design is the same regarding the geometry and the clock region organization. The outputs of these three sub-circuits go into a further neuron which provides the final output of each *macro-neuron.* Notice that there are three layers: the input layer, the output layer, and only one hidden layer.
For what concerns the input organization, the solution is rather obvious. Since the single neurons cannot evaluate nine voltages simultaneously, the three-block division of the circuit comes into help. Therefore, each block is subjected to a different triplet of inputs: the top one by the first matrix row, the middle one by the second, and the third by the last row. Ideally, using the same structure for more than nine inputs could be possible. In addition, notice that all the three neurons in each block are subjected to the same voltages. The final neural network has its basis on the just proposed circuit. Indeed, as anticipated by the scheme in the previous section, it is enough to double the structure to provide the second output. The final layout can be appreciated as a whole in figure 7.6. The circuit shows eighteen neurons in the input layer, one hidden layer of six neurons, and a two-neurons output layer. In the figure, a pair of sample outputs are also visible, highlighting that, even though the voltages to the two *macro-neurons* are the same, the outcomes could be different according to the input pattern. This behavior is due to the different molecules selection in the two *macro-neurons*.

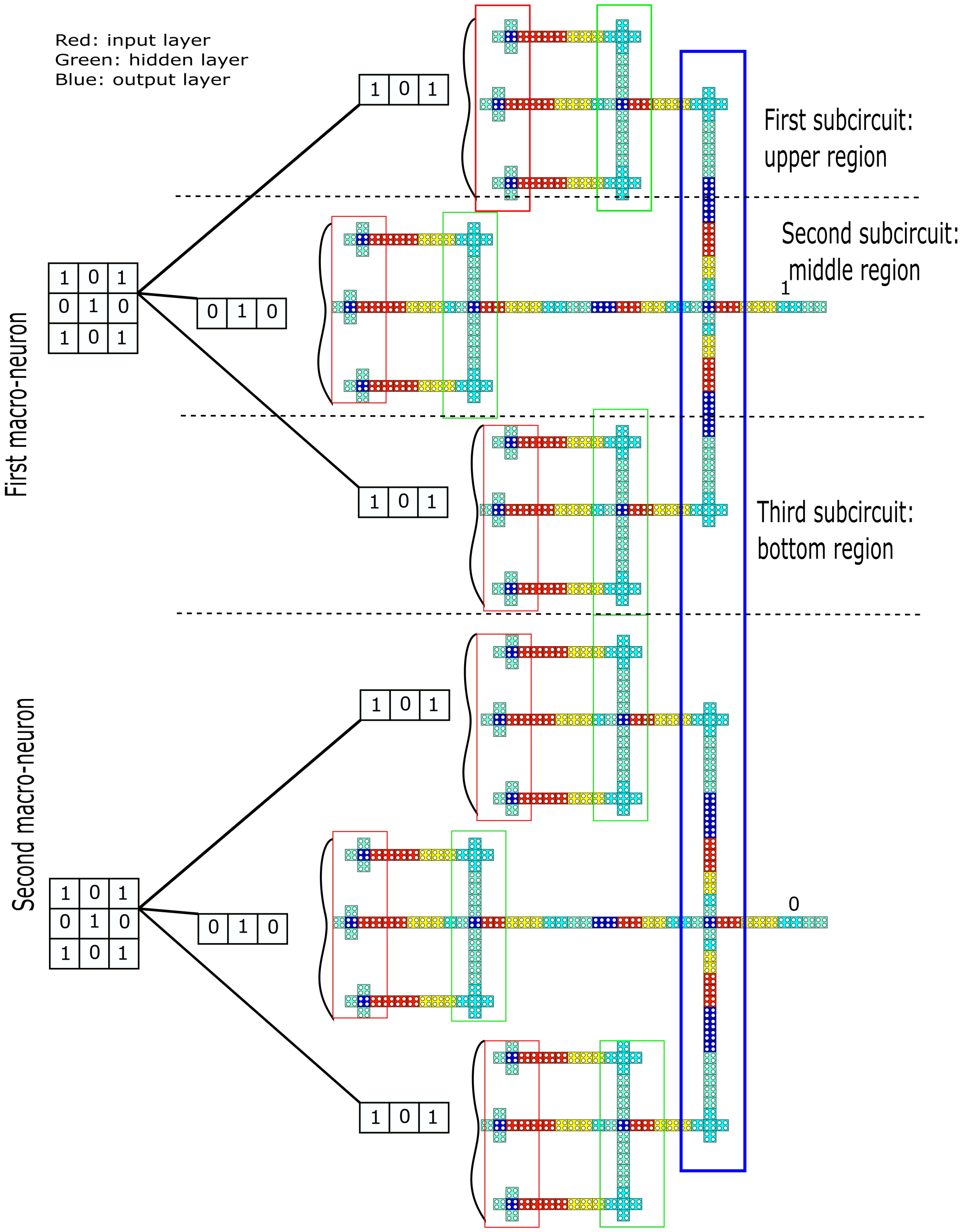


Fig. 7.6: Complete molecular structure schematic

7.3.2 Training phase and molecules selection

Designers can provide the training phase of hardware neural networks in different ways, as exhaustively explained in the second chapter of this thesis. However, a trial and error solution was preferred in this first molecular FCN neural network analysis. This choice is

mainly due to the need for grounding rules combined with the simplicity of the proposed design. Another possible solution consists of a weight coefficient normalization of the numbers provided by the software training phase.
Focusing on the molecular FCN layout design, consider Pattern A. The outputs must be, respectively for *macro-neuron* 1 and 2, logic '1' and '0'. Concerning the first, it is possible, in an initial approximation, to place the interface molecules in such a way to have logic '1' at the output of each neuron of the input layer. According to that, consider two possible neuron circuit layouts:

- **Inverting layout**: in this case interface m3 has $\alpha = 1V$, while the other two are characterized by $\alpha = 3V$. This layout ensures the less present input on m3 to be propagated
- **Non-inverting layout**: in this second solution interface m3 has α value equal to 3V, and m1 = m2 have $\alpha = 1V$. In this way, the propagation of the more present inputs is ensured.

So, for what concern Pattern A in *macro-neuron* 1, it is enough to place non-inverting layouts in the top and the bottom neurons and inverting arrangements in the middle. In this way, the outputs of each neuron will be logical '1', and it is enough to place at the output neuron interface molecules having $\alpha = 1V$ to ensure the correct behavior. This straightforward solution works well for Pattern A, Pattern B, and Pattern C.
Some problems may arise concerning Pattern D. Indeed, the desired output would be equal to '0', whereas the proposed solution would provide a '1', thus leading to a classification error. Therefore, some further reasoning is needed.
Consider Pattern A and Pattern D. The primary difference between those two is in the first input row. The computation in *macro-neuron* 1 provides a logic '1' and a '0', respectively for the two cases. These are precisely the values desired in output for correct classification. For this reason, it is necessary to change the interfaces at the output neuron such that the input coming from interface m1 will be almost automatically transported to the output. To do that, the output neuron changes as follows:

- **interface m1**: $\alpha = 1V$
- **interface m2**: $\alpha = 2.5V$
- **interface m3**: $\alpha = 3V$

Notice that for the two non-mentioned patterns, the results will be correct. Indeed, according to the same reasons, the output of the circuit will be the one coming from the upper side of the network, which is congruent to the expected result.
For what concerns the second *macro-neuron* it is interesting to notice that symmetric reasoning can be provided. Indeed, the behavior of the circuit subparts is simply reversed from the previous case. According to that, the layout of this second sub-circuit is opposite to the other. The interfaces design choices for the bottom *macro-neuron* are listed here below:

- **interface m1**: $\alpha = 2.5V$

- **interface m2**: $\alpha = 1V$
- **interface m3**: $\alpha = 3V$

In this case, interface m2 will provide the output-driving voltage input.
The complete layout selections for *macro-neuron 1* and *macro-neuron 2* are listed in the following. It is important to remark the symmetric properties of the two developed layouts. For the upper *macro-neuron* the final chosen solution shows:

- **Input layer neurons**
  - neurons in the upper and bottom parts of the circuit
    - **Interface m3**: α = 3V
    - **Interface m2=interface m1**: α = 1V
  - middle-circuit neuron
    - **Interface m3**: α = 1V
    - **Interface m2=interface m1**: α = 3V
- **Hidden layer neurons**
  - all the interface molecules characterized by $\alpha = 1V$
- **Output layer neuron**
  - **Interface m3**: α = 3V
  - **Interface m2**: α = 2.5V
  - **Interface m1**: α = 1V

On the other hand, the bottom *macro-neuron* has the following characteristics:

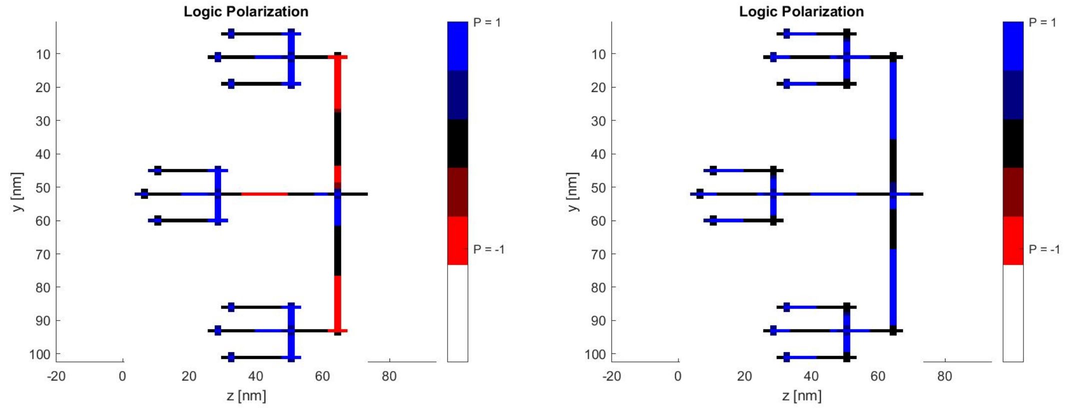
- **Input layer neurons**
  - neurons in the upper and bottom parts of the circuit
    - **Interface m3**: α = 1V
    - **Interface m2=interface m1**: α = 3V
  - middle circuit neurons
    - **Interface m3**: α = 3V
    - **Interface m2=interface m1**: α = 1V
- **Hidden layer neurons**
  - all the interface molecules characterized by $\alpha = 1V$
- **Output layer neuron**
  - **Interface m3**: α = 3V
  - **Interface m2**: α = 1V

- **Interface m1**: $\alpha = 2.5\text{V}$

The developed circuit is somehow more complex for the proposed problem than what is necessary. However, as a general rule in neural networks, a higher layer complexity would allow for evaluation errors correction and more complex task solving capabilities. Being this the first example, the choices in molecular layout and active molecules were relatively straightforward, but they can serve as a starting point for further and deeper analyses.

7.4 Graphical results for molecular FCN classifier

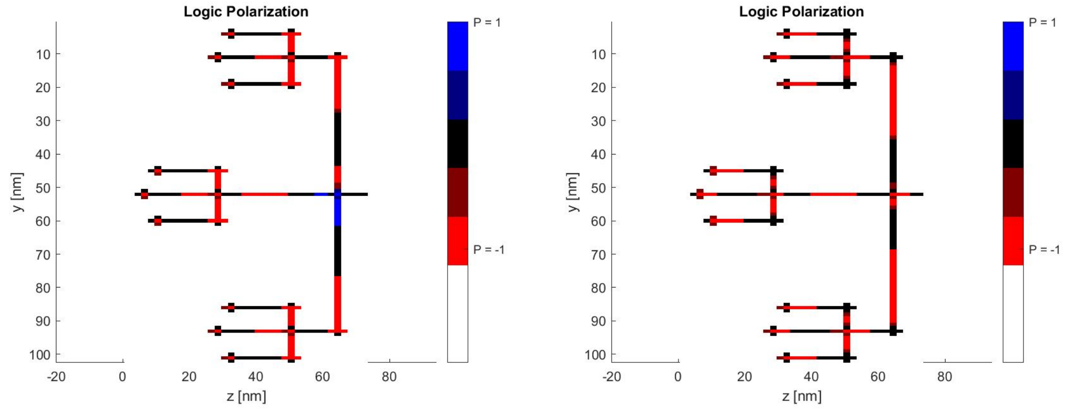
This section presents the results related to the classification problem proposed in this chapter. The two macro-neurons were simulated separately for each input pattern to avoid extremely long simulations. The figures report different instants taken in the information propagation. The logic values are encoded in terms of molecular cell polarization. In particular, a positive polarization corresponds to a high logic value, and a negative polarization to a low logic value. In the case under study, the polarization values will be either +1 or -1 in all circumstances. The clocking scheme regulating the propagation is made by five clock regions, as presented in chapter six. Of course, the main difference between the circuits analyzed in the last chapter is the higher latency. The results are presented in the following simulation snapshots. In particular, the instants reported are related to the computations in the hidden and output layers, respectively, placed in the left and right images. As it is possible to appreciate, the expected results perfectly coincide with the obtained one. This result strongly confirms the working behavior of the developed network. The classification is performed correctly for all the input patterns in the upper and bottom *macro-neurons.*



(a) *macro-neuron 1*.
Pattern A, hidden layer computation

(b) *macro-neuron 1*.
Pattern A, output layer computation

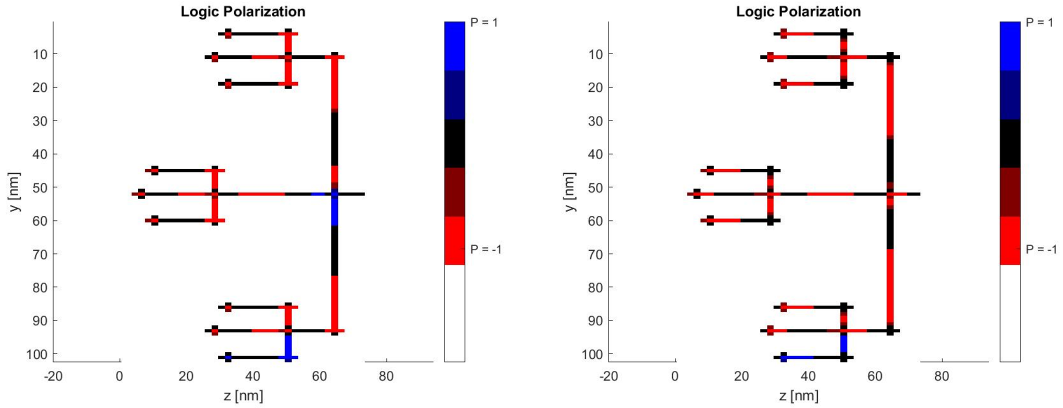
Fig. 7.7: Upper *Macro-neuron*. Pattern A evaluation. Output is logic 1



(a) *macro-neuron 2*.
Pattern A, hidden layer computation

(b) *macro-neuron 2*.
Pattern A, output layer computation

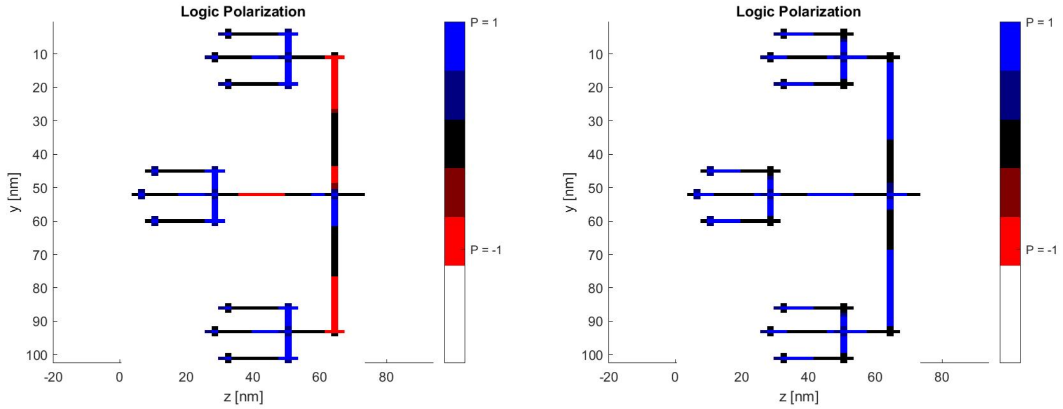
Fig. 7.8: Bottom *Macro-neuron*. Pattern A evaluation. Output is logic 0



(a) *macro-neuron 1.*
Pattern B, hidden layer computation

(b) *macro-neuron 1.*
Pattern B, output layer computation

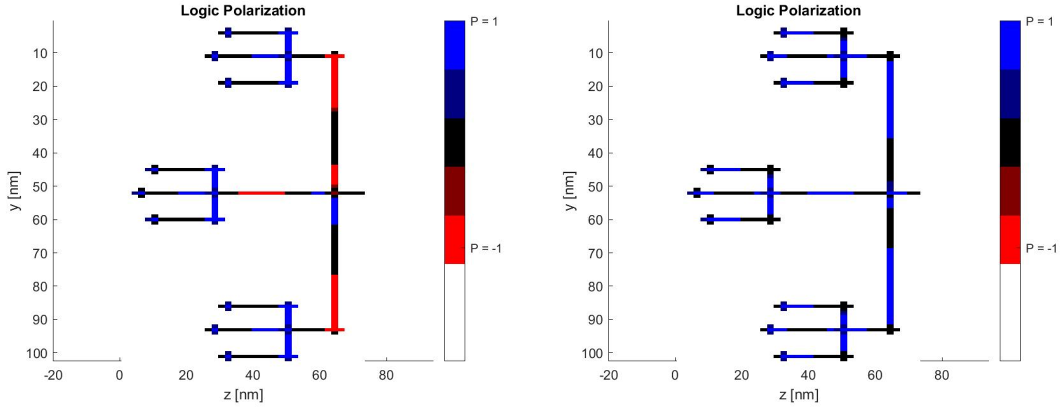
Fig. 7.9: Upper *Macro-neuron.* Pattern B evaluation. Output is logic 0



(a) *macro-neuron 2.*
Pattern B, hidden layer computation

(b) *macro-neuron 2.*
Pattern B, output layer computation

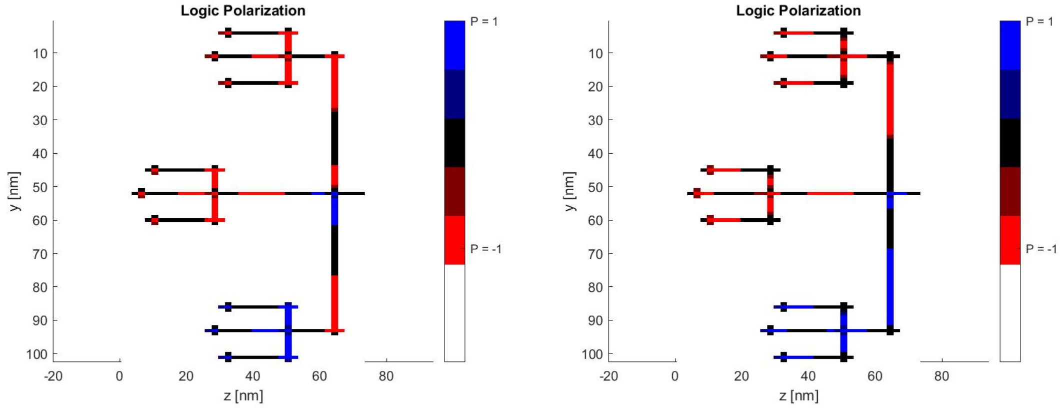
Fig. 7.10: Bottom *Macro-neuron.* Pattern B evaluation. Output is logic 1



(a) *macro-neuron 1*.
Pattern C, hidden layer computation

(b) *macro-neuron 1*.
Pattern C, output layer computation

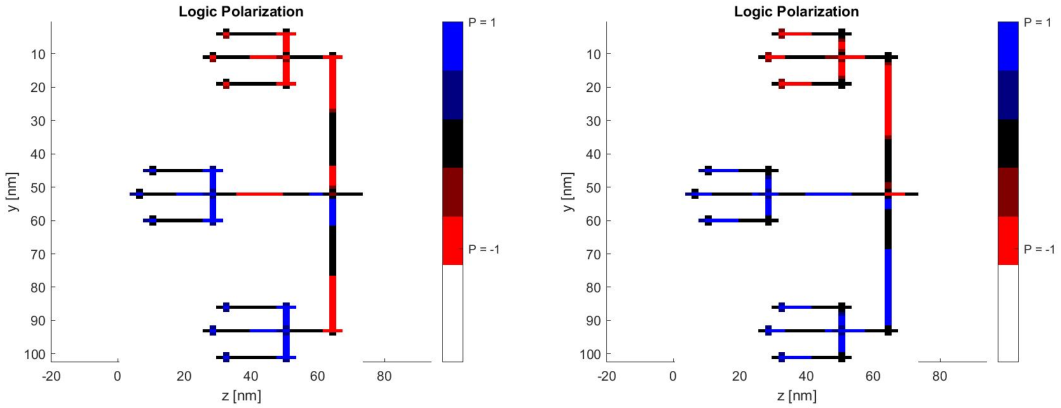
Fig. 7.11: Upper *Macro-neuron*. Pattern C evaluation. Output is logic 1



(a) *macro-neuron 2*.
Pattern C, hidden layer computation

(b) *macro-neuron 2*.
Pattern C, output layer computation

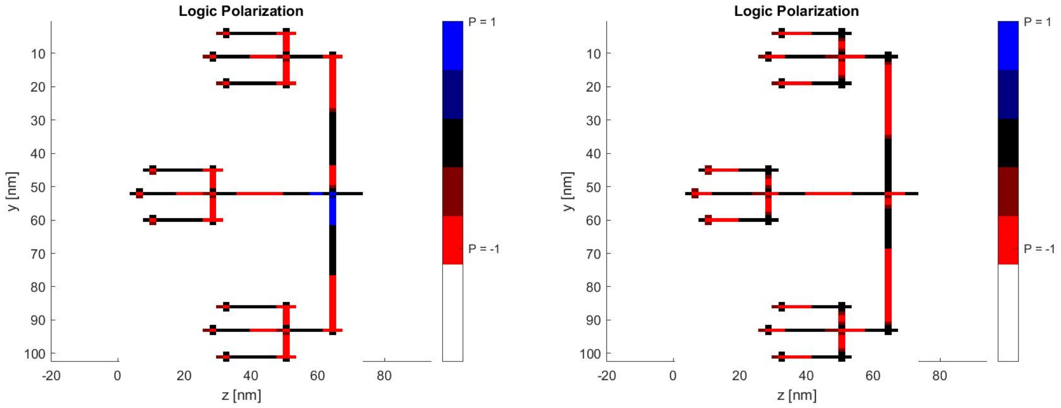
Fig. 7.12: Bottom *Macro-neuron*. Pattern C evaluation. Output is logic 1



(a) *macro-neuron 1.*
Pattern D, hidden layer computation

(b) *macro-neuron 1.*
Pattern D, output layer computation

Fig. 7.13: Upper *Macro-neuron.* Pattern D evaluation. Output is logic 0



(a) *macro-neuron 2.*
Pattern D, hidden layer computation

(b) *macro-neuron 2.*
Pattern D, output layer computation

Fig. 7.14: Bottom *Macro-neuron.* Pattern D evaluation. Output is logic 0

7.4.1 Robustness analysis of the proposed network

In designing any neural network, it is crucial to understand and mathematically evaluate the correctness rate of the outputs. This procedure can be done for variations of the original input patterns. This section provides a robustness analysis of the proposed neural network. Specifically, different patterns were presented at the circuit inputs. The new 3x3 matrices are modifications of the original Pattern A, shown in the previous section. A perfectly working classifier will classify all the new inputs as '00' or '11'. If the circuit provides the same output for a particular modified pattern as for *Pattern A* or *Pattern*

B, it is possible to state that the network is not trained enough to recognize that specific input pattern as wrong. The designer's goal is to minimize these events, meaning that the choice of the weights must be the best one. Notice that in this section are presented just the result of this kind of analysis without adjusting the weight coefficients, i.e., the molecules. This way, it is possible to demonstrate the quality of the proposed solution.

Modified input patterns and numerical results

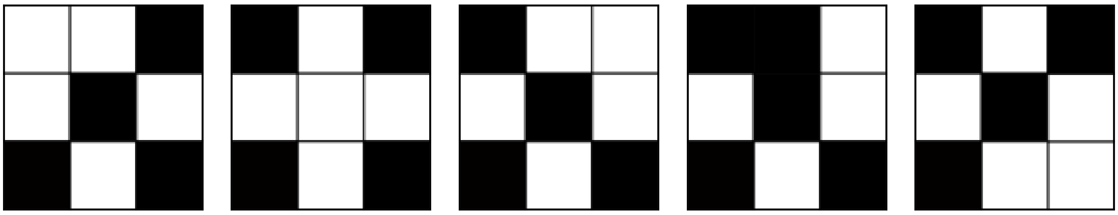
Several trials must be performed to provide a valuable analysis of input error rejections. Notice that *Pattern C* and *Pattern D*, can be already seen as variations of *Pattern A* and, consequently, classified as '11' and '00', respectively. This study aims to enlarge the concept by introducing several wrong input patterns starting from *Pattern A*. The variations consist of one or more pixels switching their content to mimic the incoming wrong values to the circuit inputs.
In figures 7.15a-7.16h, are reported the new patterns introduced to perform the analysis. Table 7.2 reports the outcomes of the circuit when subjected to each of these new patterns. There are three different types of results. The first one is related mainly to *Pattern A* and *Pattern B*, which must be classified as '10' and '01', respectively. Indeed, in the previous section, has been already demonstrated the functionality of the network in this sense. Then, any input modification of the two main patterns should be classified in another way, namely '00' or '11'. However, for some input combinations, the circuit cannot provide the correct classification and recognizes the pattern as a correct one. These cases should represent the least frequent condition for a classifier. In table 7.2, each input set is associated with the provided classification and its correctness. In addition to this, it is important to specify that the *Network interpretation* column presents how the circuit interprets the pattern, i.e., correct or wrong. Then, the *Classification correctness* is the quality of the proposed classification. Indeed, even though the network interprets a certain pattern as correct, the result can be wrong due to a non-correct interpretation. In such a case, the training phase should be repeated to refine and improve the classification to overcome the mentioned issue.
Some analysis of the results is needed at this point. It is important to recall the final decisions taken for the molecules at the various interfaces of the neural network. Consider, for instance, the pattern labeled with the letter E. The outcome of the neural network is '00', meaning that the network can classify this input set as wrong or, at least, different either from *Pattern A* and *Pattern B*. This result is mainly due to the presence of two logic '0' in the first row of the matrix. So, the output of *macro-neuron 1* is a '0', following the outcome of the upper inner layer coming from interface m1 connected to the output layer. The same considerations can be provided for *Pattern G*. A different case can be highlighted, for instance, for *Pattern F*. In that situation, the modification to *Pattern A* is in the central pixel in the middle row. However, the middle row is almost non-influencing the computation on the output layer according to the chosen weights. Indeed, interface m3 connected to the output neuron has molecules characterized by $\alpha = 3V$. The situation just commented can be considered the first problem related to the designed network: any variation in the middle row is not taken into account by the network. Concerning *macro-neuron 2*, the driving interface at the output neuron site is m2. On the bottom neuron

of the circuit, the stronger interface is, instead, m3, on which molecules with $\alpha = 1V$ are placed. Taking *Pattern L* as an example, it is, therefore, clear the reason why it is classified as not correct.

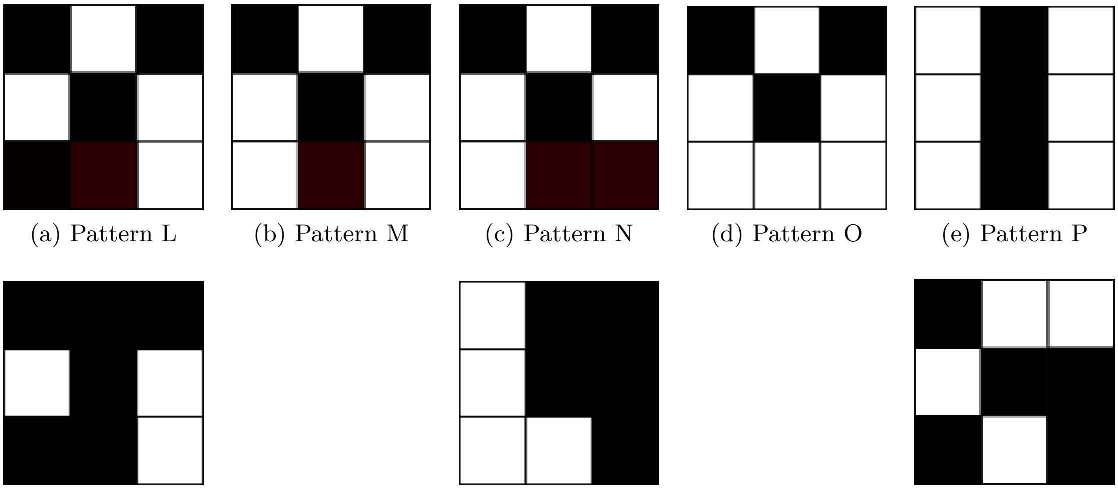
Summing up all these considerations, it is possible to conclude that the network is not reliable for variations in the middle row of the 3x3 pixel matrix. The first and third rows' most influential interfaces at the input layer side are m1 and m3. For this reason, the network recognizes as wrong all the patterns with different configurations in those pixels with respect to patterns A and B. However, for some input combinations, the network as it is is not able to guarantee correct classification. This problem is related to the choice of the weights. Indeed, the most probable way to solve it is to increase the complexity of the network by selecting different molecules for the hidden layer and, parallel to that, find a solution to improve the evaluation at the input and output layers. A possible proposal involves using other molecules to those adopted in this network, exploring solutions for α values different from 1V and 3V. This way, it would be possible to explore better the computation capability of the proposed network layout.

Correcteness rate evaluation

Through the simulations carried on and discussed in this chapter, it is possible to evaluate the rate at which the designed network provides a proper and correct classification of the pattern at its inputs. From table 7.2, it is possible to evaluate this number. Indeed, correctly classifying eleven patterns among the overall seventeen analyzed makes the correctness rate equal to 64.7%. Although this value represents a good starting point for a molecular FCN neural network, it should be increased in future works. Moreover, it is also possible to make an estimation considering a higher number of possible pixel combinations in the 3x3 input matrix and considering the network analysis. From the by-hand evaluation of 226 different arrangements of the 3x3 pixel matrix, it was calculated that the rate at which the output should be correct is 63%. This value is highly approximated but confirms the rate evaluated from the simulation outcomes. Once again, it is important to remark that, despite this being a good starting point, future analyses should guarantee better results.



(a) Pattern E (b) Pattern F (c) Pattern G (d) Pattern H (e) Pattern I



(a) Pattern L (b) Pattern M (c) Pattern N (d) Pattern O (e) Pattern P

(f) Pattern Q (g) Pattern R (h) Pattern S

Fig. 7.16: 3x3 binary images obtained by modifications of *Pattern A* and *Pattern B*

Pattern	Output	Network interpretation	Classification correctness
A	10	correct	✓
B	01	correct	✓
C	11	correct	✓
D	00	correct	✓
E	00	wrong	✓
F	10	correct	✗
G	00	wrong	✓
H	10	correct	✗
I	10	correct	✗
L	11	wrong	✓
M	11	wrong	✓
N	11	wrong	✓
O	10	correct	✗
P	01	correct	✗
Q	11	wrong	✓
R	10	correct	✗
S	00	wrong	✓

Table 7.2: Results of the robustness analysis

Chapter 8

Conclusions

The subjects treated in this work include several aspects of technology. Nowadays, the demand for neural network applications is enormous and, in some cases, moves far from the classical computing problems. The challenges related to hardware neural networks are still a lot. Parallel to these, there is the need to find solutions to overcome the issues associated with CMOS scaling. According to that, research is ongoing in different fields. In particular, FCN has proven to be one of the most exciting and innovative solutions to improve device speed and power consumption. The main advantage related to this technology is the absence of current flow. In this work, starting from the intrinsic characteristics of the molecules involved, FCN neuromorphic computation has been explored. This thesis designs a working solution for neural networks relying on molecular Field-Coupled Nanocomputing. The main challenges were the design of a proper single neuron cell and the transport of the information through the structure. Related to this second task, the main challenge was properly designing the clock region distribution and timing, considering the different molecules in the circuit layout.

The various parts have been analyzed separately. First, the focus was on the design of the neuron and the verification of the proper behavior compared to the state-of-the-art neuromorphic computation. It has been shown that the neuron behavior is coherent and predictable for different combinations of interface molecules and input voltages. Moreover, the definition of dark regions in which the neuron behavior cannot be predicted has also been analyzed in detail.

In the second part of the work, the goal was to define a propagation structure capable of transporting the information from one neuron to another without loss of information. In that study, saturator molecules were introduced as a valuable solution to ensure stable and fully digital information propagation. Furthermore, the clock signals profile has been defined considering the molecules present in the layout. The solution ensures correct propagation and information transport in the different parts of the final structure.

A step-by-step procedure was adopted to build a first working prototype of a molecular FCN neural network. The final solution involved a total of four neurons together. The network was in-depth analyzed to explore the functionality and find solutions to ensure the correct working behavior of the circuit. All the results were compared to those obtained by hand calculations using the weighted sum of the input voltages. This thesis defines possible critical situations that can arise, considering the molecules proposed so far. For this purpose, a new molecular neuron model was proposed to account for diagonal and spurious contributions to each interface polarization, with the possibility of enlarging the prediction capabilities of the circuit behavior.

The network was then arranged to deal with pattern recognition problems. Precisely, 3x3 matrix patterns form the inputs to the neural network. With the solution proposed it was possible to reach classification correctness up to 64%.

The aspects that should still be analyzed are many. One of the most important ones consists of enlarging the neural model, starting from the solution proposed in this thesis, to improve the prediction coherency to the circuit outcomes. Starting from this point, building more complex networks capable of solving more complex tasks would be possible. In parallel to this, new structures for propagation should be studied to increase the network connectivity. This way, the network would become more complex, increasing its flexibility and solving properties. In conclusion, it is worth mentioning the necessity of reliable techniques to evaluate the energy consumption of the molecular neural networks to properly compare the results with the dissipation values characterizing the state-of-the-art silicon-based neural networks.

Bibliografia

[1] G.E.Moore, “Cramming more components onto integrated circuits,” *Electronics*, 1965.

[2] “Cramming more components onto integrated circuits.” https://www.computerhistory.org/collections/catalog/102770822#. Accessed: 2022-03-29.

[3] Bernstein, C. III, Porod, Seabaugh, and Welser, “Device and architecture outlook for beyond cmos switches,” *Proceedins of the IEEE*, 2010.

[4] Nikonov, “Overview of beyond-cmos devices and a uniform methodology for their benchmarking,” *Proceedings of the IEEE*, 2012.

[5] R. W. Keyes, “Fundamental limits of silicon technology,” *Proceedings of the IEEE*, vol. 89, no. 3, pp. 227–239, 2001.

[6] I. R. Commitee, “International technology roadmap for semiconducto4 2.0.” http://222.itrs2.net/, 2015. [Online].

[7] Blair, Yost, and Lent, “Power dissipation in clocking wires for clocked molecular quantum-dot cellular automata,” *Journal of Computational Electronics*, vol. 9, pp. 49–55, 2010.

[8] Timier and Lent, “Power gain and dissipation in quantum-dot cellular automata,” *Journal of applied physics*, vol. 91, p. 823, 2002.

[9] Lent, Isaksen, and Lieberman, “Realization of a functional cell for quantum dot cellular automata,” *Science*, vol. 277, pp. 928–930, 1997.

[10] Orlow, Imre, Csaba, Li, Porod, and Bernstein, “Magnetic quantum-dot cellular automata. recent developments and prospects,” *Journal of Nanoelectronics and Optoelectronics*, vol. 3, no. 1, pp. 55–68, 2008.

[11] Lu and Lent, “Theoretical study of molecular quantum-dot cellular automata,” *Journal of computational electronics*, vol. 4, pp. 115–118, 2005.

[12] Lent, Isaksen, and Lieberman, “Molecular quantum-dot cellular automata,” *Journal of American Chemical society*, 2002.

[13] Srivastava and Sarkar, "Estimation of upper bound of power dissipation in qca circuits," *IEEE transactions on nanotechnology*, vol. 8, pp. 116–127, 2009.

[14] Ardesi, Turvani, Graziano, and Piccinini, "Scerpa simulation of clocked molecular field coupling nanocomputing," *IEEE transactions on Very Large Scale Integration*, vol. 30, 2015.

[15] Tòth and Lent, "Quasiadiabatic switching for metal-island quantum-dot cellular automata," *Journal of applied Physics*, vol. 85, no. 5, pp. 2977–2984, 1999.

[16] Cowburn and Welland, "Room temperature magnetic quantum cellular automata," *Science*, vol. 287, no. 5457, pp. 1466–1468, 2000.

[17] Governale, Macucci, Iannone, and Ungarelli, "Modelling and manufacturability assessment of bistable quantum-dot cells," *Journal of applied Physics*, vol. 85, no. 5, pp. 2962–2971, 1999.

[18] Wang and Lieberman, "Thermodynamic behavior of molecular-scale quantum dot cellular automata wires and logic devices," *IEEE Transactions on nanotechnologies*, vol. 3, no. 3, pp. 368–376, 2004.

[19] Arima, Iurlo, Zoli, Kumar, Piacenza, D. Sala, Matino, Maruccio, Rinaldi, Paolucci, Marcaccio, Cozzi, and Bramanti, "Toward quantum-dot cellular automata units: Thiolated-carbazole linked bisferrocenes," *Nanoscale*, vol. 4, pp. 813–823, 2012.

[20] Lent, Liu, and Lu, "Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling," *Nanotechnology*, vol. 17, pp. 4240–4251, 2006.

[21] Tsukerblat, Palii, Clemente, Suaud, and Coronado, "Quantum cellular automata: a short overview of a molecular problem," *Proceedings of the European Conference Physics of magnetism*, vol. 133, no. 3, pp. 329–335, 2018.

[22] Pulimeno, Graziano, Antidormi, Wang, Zahir, and Piccinini, "Understanding a bisferrocene molecular qca wire," *Springer Berlin Heidelberg*, 2015.

[23] Pulimeno, Graziano, Sanginario, Cauda, Demarchi, and Piccinini, "Bis-ferrocene molecular qca wire: ab inition simulations of fabricarion driven fault tolerance," *IEEE Transactions on Nanotechnology*, vol. 12, no. 4, pp. 498–507, 2013.

[24] L. Zoli, *Active bis-ferrocene molecules as unit for molecular computation.* PhD thesis, Università di Bologna, 2009.

[25] Lent and Isaksen, "Clocked molecular quantum dot cellular automata," *IEEE Transactions on Electron Devices*, vol. 50, no. 9, pp. 1890–1896, 2003.

[26] Wang, Chilla, Palucci, Graziano, and Piccinini, "An effective algorithm for clocked field-coupled nanocomputing paradigm," *Proc. Nanotechnol. Mater. Devices Conf (NMDC)*, pp. 1–2, 2016.

[27] Ardesi, Wang, Turvani, Piccinini, and 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 intergrated circuits and systems*, vol. 39, no. 10, pp. 2749–2761, 2020.

[28] M. F. et al., "Gaussian 09 revision a.1," Wallingford, CT, USA: Gaussian Inc., 2009.

[29] Ardesi, Pulimeno, Graziano, Riente, and Piccinini, "Effectiveness of molecules for quantum dot cellular automata," *Journal of low power electronics and applications*, 2018.

[30] Mohn, Gross, Moll, and Meyer, "Imaging the charge distribution within a single molecule," *Nature nanotechnology*, 2012.

[31] Pulimeno, Graziano, Demarchi, and Piccinini, "Towards a molecular qca wire: simulation of write-in and read-out systems," *Solid-State Electronics*, vol. 77, pp. 101–107, 2012.

[32] M. Forssel, "Hardware implementation of artificial neural networks," *Information flow in networks*, 2014.

[33] Hamedani, Liu, Wu, and Yi, "Reservoir computing meets smart grids: attack detection using delayed feedback networks," *IEEE Transactions on industrial informatics*, vol. 14, no. 2, pp. 734–743, 2018.

[34] Marlestone, Zamft, Maguire, Shapiro, Cybulski, Glaser, Amodei, Stranges, Kalhor, and Dalrymple, "Physical principles for scalable neural recording," *Frontiers in computational neuroscience*, vol. 7, 2013.

[35] "Electromagnetic fields at workplaces - a new scientific approach to occupational health and safety." `https://www.researchgate.net/publication/280528900`. Accessed: 2022-07-01.

[36] "Action potential - the resting membrane potential." `https://teachmephysiology.com/nervous-system/synapses/action-potential/`, 2021. Accessed: 2022-06-15.

[37] Hodgkin and Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *Journal of Physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[38] Kistler, Gerstner, and Hemmen, "Reduction of the hodgkin-huxley equations to a single.variable threshold model," *Neural computation*, vol. 9, no. 5, pp. 1015–1045, 1997.

[39] Ojha, Abraham, and Snasel, "Metaheuristic design of feedforward neural networks: a review of two decades of research," *Engineering applications of artificial intelligence*, vol. 60, pp. 97–116, 2017.

[40] Sharma, Sharma, and Athaiya, "Activation functions in neural networks," *International journal of engineering applied sciences and technology*, vol. 4, no. 12, pp. 310–316, 2020.

[41] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Physiological review*, vol. 65, no. 6, 1958.

[42] A. Dey, "Machine learning algorithms: a review," *International Journal of Computer Science and Information Technologies*, vol. 7, no. 3, pp. 1174–1179, 2016.

[43] Aparin and Levin, "Methods and systems for cmos implementation of neuron synapse," U.S. Patent 8,694.452 B2, Apr. 8 2014.

[44] H. Hikawa, "(fpga) implementation of self organizing map with digital phase locked loops," *Neural networks*, vol. 18, no. 56, pp. 514–522, 2005.

[45] V. Beiu and M. Avedillo, "Vlsi implementations of threshold logic - a comprehensive survey," *IEEE transactions on neural networks*, vol. 14, no. 5, pp. 1217–1245, 2003.

[46] G. Indiveri and T. Horiuchi, "Frointers in neuromorphic engineering," *Frontiers in neuroscience*, vol. 5, 2011.

[47] C. Mead and M. Ismail, "Analog vlsi implementation of neural systems," *Spinger Science and Business Media*, vol. 80, 2012.

[48] M. Kawaguchi, M. Umeno, and N. Ishii, "Analog neural circuit with switched capacitor and design of deep learning model," *3rd International conference on applied computinf and information technology/2nd International conference on Computational Science and Intelligence*, 2015.

[49] W. Liu, Z. Li, S. Xue, X. Yang, and W. Lin, "Training an artificial neural network with op-amo integrators based analog circuits," *IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*, 2018.

[50] L. Chua, "Memristor: the missing circuit element," *IEEE Transactions on circuit theory*, vol. 18, no. 5, pp. 507–519, 1971.

[51] D. Strukox, S. Gregory, and S. Duncan, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 2008.

[52] Y. Pershin and M. D. Ventra, "Experiment demonstration of associative memory with memresistive neural networks," *Neural networks*, vol. 23, no. 7, pp. 881–886, 2010.

[53] S.Ambrogio, S. Balatti, F. Nardi, and S.Facchinetti, "Spike-timing dependent plasticity in a transistor-selected resistive switching memory," *Nanotechnology*, vol. 24, no. 38, 2013.

[54] R. Douglas, M. Mahowald, and H. Wallinga, "Hybrid analog-digital architectures for neuromorphic systems," *Proceeding of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 3, pp. 1848–1853, 1994.

[55] IEEE, ed., *Neuromorphic computation using quantum-dot cellular automata*, IEEE International Conference on Reeboting Computing (ICRC), (Washington DC, USA), IEEE, 2017.

[56] G. Beretta, “Study of field-coupled nanocomputing based on molecules for neural systems,” Master’s thesis, Politecnico di Torino, Torino, Italia, Apr. 2020.

[57] Y. Ardesi, G. Beretta, M. Vacca, G. Piccinini, and M. Graziano, “Impact of molecular electrostatics on field-coupled nanocomputing and quantum-dot cellular automata circuits,” *Wlectronics*, vol. 11, no. 276, 2022.

[58] S. Bhattacharya, “An overview of neural approach on pattern recognition,” *Data Science Blogathon*, 2020.

[59] “Image recognition with deep neural networks and its use cases.” https://www.altexsoft.com/blog/image-recognition-neural-networks-use-cases/. Accessed: 2022-05-28.

[60] “Neural networks and deep learning.” http://neuralnetworksanddeeplearning.com/chap2.html. Accessed: 2022-06-02.

[61] “Deep learning toolbox.” https://it.mathworks.com/products/deep-learning.html. Accessed: 2022-06-02.

[62] G.Beretta, Y.Ardesi, and M.Graziano, “Multi-molecule field-coupled nanocomputing for the implementation of a neuron,” *IEEE Transactions on Nanotechnology*, vol. 21, pp. 52–59, 2022.