

POLITECNICO DI TORINO

Master of Science in Electronic Engineer Electronics Engineering

Master Thesis

**Modeling and Simulation of Charge  
Dynamics in Molecular Field-Coupled  
Nanocomputing Circuits**



**Supervisors**

Prof. Mariagrazia Graziano

Dr. Yuri Ardesi

M.Sc. Roberto Listo

**Candidate**

Cecilia Steconi

7 Aprile 2025

Compiled with L<sup>A</sup>T<sub>E</sub>X on March 30, 2025.

# Abstract

Molecular Field-Coupled Nanocomputing (MolFCN) stands out as a promising solution to overcome the limitations posed by CMOS scaling. MolFCN technology implements the Quantum Cell Automata (QCA) paradigm, in which the information is encoded within the so-called quantum dots of a unit cell. The information is then propagated between unit cells by electrostatic interaction.

Among beyond-CMOS technologies, MolFCN stands out for its miniaturization, low-power operation, and potential to achieve hundreds of gigahertz switching frequencies at room temperature. In MolFCN, information is encoded and propagated through electrostatic intermolecular interactions guided by a multi-phase clocking mechanism, which ensures correct signal propagation and pipelining.

The Self-Consistent Electrostatic Potential Algorithm (SCERPA) has been introduced to simulate a clocked MolFCN circuit. However, SCERPA does not include charge dynamics and its effect on information propagation and elaboration. Prior research enabled the modeling of the intramolecular charge movement using Real-Time Time-Dependent Density Functional Theory (RT-TDDFT), permitting the molecules to be approximated as first-order low-pass filters in the dynamic domain.

This thesis explores modeling the dynamic intramolecular interaction during information propagation by modeling the circuit as a chain of low-pass filters defined by molecule-specific cutoff frequencies.

The modeling and simulation are applied to butane and bis-ferrocene molecules and begin with calculating molecular chain voltages considering the electrostatic interactions. Time is discretized with the Finite Element Method (FEM), and the forward Euler method is used to solve the dynamic voltage equation, reflecting non-stationary inputs from neighboring molecules' charge redistributions. Charge distribution is determined using the Voltage-Dependent Atomic Charge Transfer (VACT) method, which links potential differences to charge profiles based on the molecule type. Multiple VACT profiles exist for bis-ferrocene, varying with clock field polarization.

The simulative setup models transient behavior and measures propagation times through molecular wires and majority voters, verifying results against expected molecular switching frequencies. Longer chains or increased molecular spacing cause signal loss, consistent with theoretical predictions.

This work extends SCERPA to enable dynamic MolFCN circuit simulations, opening new ways of simulating signal propagation dynamics in MolFCN circuits for the first time.



*A mamma e papà,  
a ZiAlma,  
ma anche un po' a me stessa.*



# Acknowledgements

Scrivo questa sezione in italiano, l'unica per ringraziare chi mi è stato a fianco per questi infiniti anni di magistrale.

Un grazie a Roberto Listo, per non aver voluto abbandonarmi alla ricerca di una nuova tesi dopo essermi presa una "pausa di riflessione" durata più un anno, senza di lui probabilmente la fine della mia carriera universitaria avrebbe dovuto aspettare ancora un po'.

Grazie a Fra, il mio amico del poli, per le mille giornate al comala a prepararci per gli esami, a perder tempo sotto il sole e ad esplorare qualsiasi down mentale il nostro cervello potesse offrire.

Un ringraziamento inaspettato al Rivarolo, squadra che ormai non sopporto, ma che si merita una menzione per avermi portata in Piemonte, per avermi portata ad iscrivermi al Poli di Torino e avermi dato la possibilità di trovarmi amicizie torinesi.

Grazie a Torino, elegante e raffinata e grazie a chi mi ha fatto da guida turistica i primi mesi. Sappi che se mi avessi raccontato storie non vere, ormai sono state tramandate a tutti i senigalliesi e non che mi hanno ascoltata ripeterle, se non ti riconosci, sei proprio una banana.

A te, Andre, per avermi sentito blaterare della tesi e su quanto fossero belli i miei grafici, per essere stato un compagno di studio nell'altra stanza: il solo fatto che lo stessi facendo anche tu mi ha dato la disciplina che mi serviva senza neanche che lo sapessi. Ti ringrazio dunque per esserci sempre stato (negli ultimi due anni e mezzo) e per avermi supportata e sopportata, senza di te e senza questi cliché questa tesi non sarebbe la stessa.

Sempre grazie alla famiglia del mulino bianco, ma soprattutto a mia mamma, per non aver smesso di litigare per farmi finire questa tesi. Telefonate rabbiose e conversazioni taglienti, ma alla fine questa tesi l'ho finita davvero.



# Contents

<b>List of Tables</b>	11
<b>List of Figures</b>	13
<b>List of acronyms and abbreviations</b>	16
<b>1 Introduction</b>	18
1.1 MolFCN - Molecular Field Coupled Nanocomputing	18
1.2 Clock introduction	19
1.3 Time-Independent Modeling MolFCN (SCERPA)	20
1.4 A first ab initio approach toward single molecule frequency analysis	20
<b>2 Toward Time-dependent Modeling</b>	23
2.1 Finite Element Method (FEM) basis	23
2.2 Modeling	26
2.3 The Voltage-Dependent Atomic Charge Transfer (VACT) function	28
2.4 Geometry of the circuits	31
<b>3 Charge dynamics Simulation algorithm</b>	35
3.1 Initialization of variables and definition of input data	35
3.2 Calculation of the voltage induced by the input driver on the molecule in the wire	36
3.3 Execution of the function that computes the temporal evolution of charge dynamics	36
3.4 Evaluation of Propagation Delay	37
3.5 Plotting of the obtained result	37
3.6 Analysis of the delay over the number of molecule, over the switching frequency and the intermolecular distance.	40
<b>4 Result of the Simulation</b>	41
4.1 Wire of molecule	41
4.2 Majority Voter	44
4.3 Study of the delay as a function of n, d and the frequency switch	46
<b>5 Conclusion and future perspectives</b>	51

<b>A</b>	<b>MATLAB code</b>	53
A.1	simulationButane.m . . . . .	53
A.2	bis_ferrocene.m . . . . .	57
A.3	Majority_voter.m . . . . .	61
A.4	EvolutionInTime.m . . . . .	65
A.5	FindDelay.m.m . . . . .	69
A.6	VACT.m and VACT_bisf.m . . . . .	70
A.7	AnalysisButane_different_d_n_f_switch.m . . . . .	72
	<b>Bibliography</b>	75

# List of Tables

2.1	Tabella dei dati con $V_{MUT}$ , $Q_1$ e $Q_2$ . . . . .	28
4.1	Reference values for simulation parameters . . . . .	46



# List of Figures

1.1	Bis-Ferrocene structure: we have the 2 ferrocene group in dot1 and dot2 in the arms of the Y, while dot 3 and the dot 4 are in the stick of the Y .	20
2.1	Electric scheme of a molecule model as an RC circuit . . . . .	23
2.2	Electric scheme of a molecule model as an RC circuit The green arrow indicates where KVL is performed 2.1, while the blue circle is in correspondence of the node where the KVI is calculated 2.2 . . . . .	24
2.3	RC response to a step input: exact solution and FEM approximation, using the switch frequency of the molecule $f_S = 640$ GHz a. Negative step input at $t = 0.5$ ps b. Exact solution of the eq. 2.4 which is an exponential decay c. The approximated solution in eq. 2.6 using FEM method is reported in blue lines, while the exact solution in the continuous red line: from this visual representation it can be said that the approximated solution doesn't introduce any major discrepancy with the exact solution . . . . .	26
2.4	VACT of the butane molecule The data point measured experimentally are highlighted, while the interpolation is shown like a continuous line. . .	29
2.5	Here reported the bis-ferrocene molecule structure, with the direction of the electric field considered for the VACT . . . . .	30
2.6	A.Clock = $-2V/nm$ If the clock is negative, molecules are forced to be stored in Q3 and Q4, so the voltage across the molecule is equal to zero. .	30
2.7	B.Clock = $0V/nm$ If the clock is not present, the transmission can happens but we will have a more non linear behaviour, enhancing the probability of mistakes in the propagation. . . . .	31
2.8	C.Clock = $+2V/nm$ With a positive clock, the VACT is almost linear. . .	31
2.9	Wire of molecule Here the wire is made by 4 molecule, each divided by one another by a distance $d = 0.6nm$ . The width of the molecule is $w = 0.6nm$	32
2.10	Space position of the molecule in a Majority Voter . . . . .	33
4.1	Evolution in time of the molecule in the molecular wire: On the right the evolution of the voltage across the molecule is represented, while on the left part the focus is made on the evolution of the distribution of charges .	41
4.2	First Butane molecule of the molecular wire with a frequency switch = $640GHz$ . . . . .	42

4.3	Last Butane molecule of the molecular wire with a frequency switch = 640GHz . . . . .	42
4.4	Evolution in time of the molecule in the molecular wire: On the right the evolution of the voltage across the molecule is represented, while on the left part the focus is made on the evolution of the distribution of charges . . . . .	43
4.5	First bisferrocene molecule of the molecular wire with a frequency switch = 250GHz . . . . .	43
4.6	Last bis-ferrocene molecule of the molecular wire with a frequency switch = 250GHz . . . . .	43
4.7	Last molecule of the molecular wire with negative clock: charge distributions of Q1 and Q2 are fixed to zero since the whole distribution is in the other dot. Hence, the molecule is in a null configuration . . . . .	43
4.8	Molecular wire without clock . . . . .	44
4.9	Last 2 molecules of the Majority Voter, forming an unique QCA . . . . .	45
4.10	A . . . . .	45
4.11	B . . . . .	45
4.12	A.Last molecule of the Majority Voter with Butane molecule, calculated with a $f_S = 650$ GHz, with a resulting delay = 1.2954 ps $\hat{=}$ 1.3 ps, while on B we can observe the input, the output associated to the configuration shows in A and the exact delay . . . . .	45
4.13	A . . . . .	46
4.14	B . . . . .	46
4.15	A. Last molecule of the Majority Voter with Bis-Ferrocene molecule, calculated with a $f_S = 250$ GHz, performed in the case where 2 input are the ones showed in B, along with the final Output (denoted as last) and the exact delay. . . . .	46
4.16	Delay as a function of the switch frequency of the molecule . . . . .	47
4.17	Delay as a function of the number of molecule that composed the molecular wire . . . . .	47
4.18	Delay as a function of the number of molecule with the opposite input . . . . .	48
4.19	Delay as a function of the intermolecular distance . . . . .	48
4.20	Charge distribution when d is too low. . . . .	49
4.21	Charge distribution when d is too high. . . . .	49



# List of acronyms and abbreviations

<b>MolFCN</b>	molecular Field Coupled Nanocomputing
<b>FCN</b>	Field Coupled Nanocomputing
<b>ITRS</b>	International Technology Roadmap for Semiconductors
<b>CMOS</b>	Complementary Metal-Oxide-Semiconductor
<b>MOSFET</b>	Metal-Oxide-Semiconductor Field-Effect Transistor
<b>VACT</b>	Voltage-Dependent Atomic Charge Transfer
<b>MUT</b>	Molecule Under Test
<b>SCERPA</b>	Self-Consistent Electrostatic Potential Algorithm
<b>QCA</b>	Quantum Cell Automa
<b>KVL</b>	Kirchoff's Voltage Law
<b>KCL</b>	Kirchoff's Current Law
<b>FEM</b>	Finite Element Method



# Chapter 1

## Introduction

The enhancement of digital electronics performance in recent decades has been driven primarily by the scaling of CMOS (Complementary Metal-Oxide-Semiconductor) technology. This technology has served as the foundational framework for modern electronics, facilitating the development of smaller and faster devices. CMOS transistors are the essential components of most integrated circuits (ICs) used in a diverse range of applications, from smartphones to supercomputers.

Currently, transistor dimensions are near to the nanometric scale, raising concerns about the limitations of further scaling. Historically, the reduction of transistor size has been associated with an increase of the transistor density, resulting in an improved device performance. However, following the road of further miniaturization introduces significant challenges, including quantum tunneling, heat dissipation, and leakage currents. These issues diminish the advantages of scaling and indicate a gradual decline in the relevance of Moore's Law, as the process of shrinking transistors becomes progressively more complex and expensive, with no guarantee of improvement on the final performance. [1]

In response to these challenges, the industry is actively investigating alternative technologies to extend computing power beyond the constraints of CMOS scaling. This field, referred to as "beyond-CMOS", is focused on developing new computing paradigms based on innovative physical phenomena. One particularly promising approach is the use of Molecular Field-Coupled Nanocomputing (MolFCN) technique, which enables the transmission of information without the flow of current, thereby significantly reducing power consumption. This can be reached thanks to the use of molecule instead of transistor, encoding binary information on the charge distribution of polarized molecule.

### 1.1 MolFCN - Molecular Field Coupled Nanocomputing

Molecular Field-Coupled Nanocomputing (MolFCN) leverages electrostatic interactions, such as dipole-dipole coupling and Coulomb forces, to perform computations at the molecular scale. In this approach, molecules act as information storage units, where data is

encoded in the spatial arrangement and interactions of charges within nanoscopic structures. The distribution of these charges or dipoles is influenced by external electric fields, facilitating efficient information transfer between adjacent elements, similar to signal propagation in conventional circuits.

To exploit these properties for computation, research focuses on molecules with a high susceptibility to polarization, like the butane or the bis-ferrocene, which will be used in this work. These molecules exhibit bistability, meaning that they can adopt two stable configurations, each corresponding to a distinct binary state. This characteristic enables reliable data storage and processing at the molecular level. To obtain an even better stabilization of the information, two molecules are usually considered as an unitary cell, also known as the Quantum Cellular Automata paradigm.

The QCA is a square-shaped cell composed of six quantum dots with two mobile charges. Since molecule like butane are characterized by 2 possible centers of charge aggregations, two nearby molecule can be depicted as a QCA. Molecular FCN used this configuration as a basic element for electronic circuits [2]. In QCA, molecules are characterized by charge aggregations points, playing the role of QCA 'dots'. This happens thanks to the redox centers of the molecule which favour the aggregation of the electron in precise regions. Binary information can be encoded in the dots of the molecule, opening the possibilities to build molecular wires and logic gates. The fundamental principle that allow the transmission of the information is the principle of lowest potential energy, which states that physically any natural system will evolve toward the configuration with the lowest potential energy, hence equal charge will tend to repulse themselves. [2]

## 1.2 Clock introduction

The clock field is an additional electric field introduced to create multi-phase circuits with an orthogonal direction with respect to the direction of propagation. This field can enhance the switching ability of the cell or confine the charge in the null state, preventing the molecule from storing information. This is useful because in long circuit there is high possibilities of logic errors during the propagation of the information along the whole chain. With the use of the clock field it is possible to divide the circuit, enabling the switching activities only for a small portion of the circuit at a time with a consequent increase on the probability of a correct transmission. [3] For the use of the clock, molecule needs more than 2 possible centers of aggregation for the charges, denoted as 'dot' of the molecule, like the bis-ferrocene molecule. The bis-ferrocene in fact is composed of 4 'dot', 2 of them will lead to a polarized molecule, the other 2 will confined the charge in the null-state.

In 1.1 it is reported the bisferrocene molecule, which has a Y-shaped molecule. The charge aggregation point are also highlighted

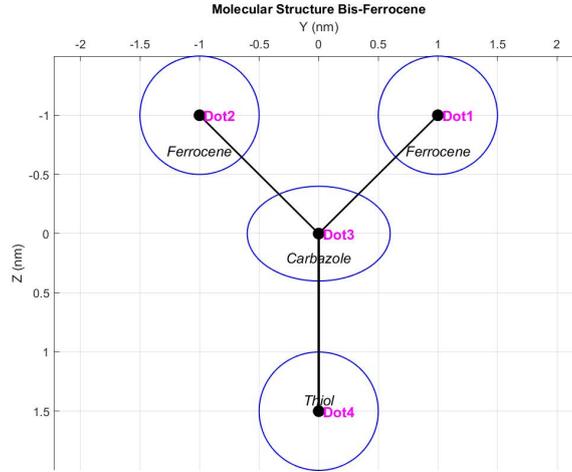


Figure 1.1: Bis-Ferrocene structure: we have the 2 ferrocene group in dot1 and dot2 in the arms of the Y, while dot 3 and the dot 4 are in the stick of the Y

### 1.3 Time-Independent Modeling MolFCN (SCERPA)

SCERPA stands for Self-Consistent Electrostatic Potential Algorithm, it is a computational method that proposes a simulation of molecular wire in its stationary behaviour. The algorithm proposes an iterative evaluation of the voltage across a molecule and the use of the Voltage-Dependent Atomic Charge Transfer (VACT) to retrieve the molecule charge distribution. SCERPA estimates the electrostatic potential using an aggregated charge model, approximating the behavior of an isolated molecule. This assumption significantly reduces computational costs. However, it also prevents SCERPA from accounting for hindrance effects, introducing negligible minor errors[2]. The SCERPA algorithm is currently implemented in MATLAB, establishing the foundation for analyzing MolFCN circuits. As previously mentioned, SCERPA skipped ab initio molecular simulations, considering the molecule only as a charge distributions aggregated in well-defined region denoted as 'dot'. These charge distributions generate an electric field, which subsequently influence the distributions on neighboring molecules. The strength of this influence depends on the distance between molecules, following electrostatic theory. SCERPA applies this principle iteratively across the entire molecular circuit, effectively capturing the interactions and their cumulative effects [4].

### 1.4 A first ab initio approach toward single molecule frequency analysis

The state of the art has already studied how the molecule reacts to an external input, finding a dynamic characterization of the molecule. The molecule under studies was subjected to time-varying electric fields in order to analyze the resulting dipole moment

variation and searching for its limit in terms of switching frequency. The found switching frequency for the butane molecule is of 640 GHz, which will be used as the switching frequency in our simulations. Another interesting result coming from previous work is that the molecule has a single pole transfer function. This knowledge, will be the one of the fundamental assumption on which this work is built on, since this knowledge will be used for creating an equivalent circuit which is characterized by a single pole transfer function, hence an RC circuit. [5]



## Chapter 2

# Toward Time-dependent Modeling

### 2.1 Finite Element Method (FEM) basis

As it was stated previously, recent studies have demonstrated that molecules can be modeled as low-pass filters with dedicated cutoff frequency for every molecule. Since the electrical response of a first-order low-pass filter can be represented as an RC circuit, the molecule itself can also be effectively modeled in this way.

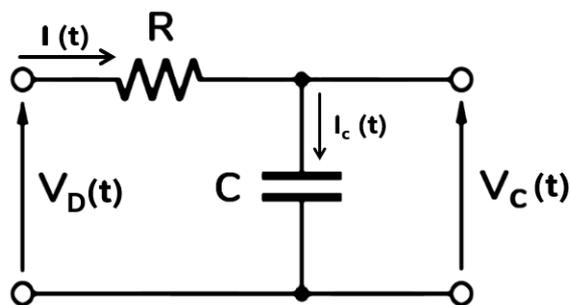


Figure 2.1: Electric scheme of a molecule model as an RC circuit

It is possible to use pic.2.1 as the equivalent circuit of the molecule, where  $V_C$  represents the voltage across the molecule and  $V_D$  serves as the input voltage. By modeling the molecule as an electronic circuit, fundamental electronic laws can be applied to its equivalent circuit, like the Kirchhoff's Laws and the capacitor equation.

$$V_D(t) = R \cdot I(t) + V_C(t) \quad (2.1)$$

$$I(t) = I_C(t) \quad (2.2)$$

$$I_C(t) = C \cdot \frac{\delta V_C}{\delta t} \quad (2.3)$$

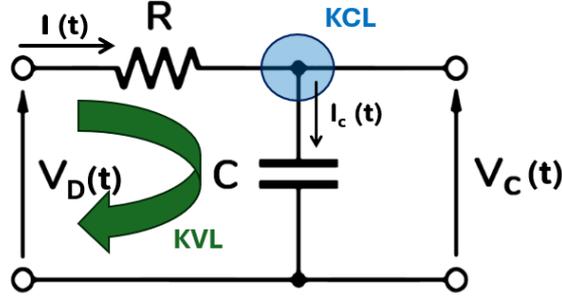


Figure 2.2: Electric scheme of a molecule model as an RC circuit  
 The green arrow indicates where KVL is performed 2.1, while the blue circle is in correspondence of the node where the KVI is calculated 2.2

By applying Kirchhoff's Current Law (eq. 2.2), the capacitor current  $I_C$  from eq. 2.3 can be used to substitute the current  $I(t)$  in Kirchhoff's Voltage Law (eq. 2.1). Assuming  $\tau = RC$  the following differential equation is obtained (eq. 2.4) :

$$V_D(t) = \tau \cdot \frac{\delta V_C}{\delta t} + V_C(t) \quad (2.4)$$

If we consider  $V_D(t)$  as a step function  $u(t)$  we can reconduce 2.4 to the following form:

$$A \frac{\delta f(t)}{\delta t} + Bf(t) = u(t),$$

which has a well known solution: the exponential function, which corresponds, in fact, to the transfer function of the RC circuit. Since the input of the molecular wire can be seen as a step function with an amplitude equal to the voltage across the molecule driver, the expected output of the molecule will be itself an exponential function.

In this work, time is discretized for computational purposes. Therefore, while the exact analytical solution serves as an initial reference, an approximated numerical solution is employed to simulate the system dynamically. The discretization of time is necessary because, in practical simulations, an infinite number of time steps cannot be computed. Instead, a finite number of steps are possible to be elaborated, where each time step  $\delta t$  is chosen to be significantly smaller than the characteristic switching speed of the molecule. Under this condition, each step can be considered a small incremental evolution of the system, ensuring an accurate numerical representation of the transient response.

To achieve this, the Finite Element Method (FEM) is applied. FEM is a widely used numerical technique for solving differential equations, particularly in physics and engineering, where systems are too complex for analytical solutions. Instead of solving the entire equation at once, FEM discretizes the time domain into smaller subintervals and solves the governing equation iteratively over each interval. This approach transforms the

continuous problem into a discrete set of algebraic equations, making it computationally feasible.

So in the equation 2.4  $\delta t$  can be substitute with a finite time step  $\Delta t$ :

$$V_D(t) = \tau \cdot \frac{\Delta V_C}{\Delta t} + V_C(t) \quad (2.5)$$

There are different possibilities for the FEM solution of differential equations. In this work, the explicit method denoted as forward Euler is applied. Firstly, the eq. 2.5 shall be written in the form

$$V_C(t + \Delta t) = V_C(t) + V_D(t)$$

where the function at  $t = t + \Delta t$  is in the left part of the equation, leaving the function at  $t = t$  in the right one. Writing the equation in this way, it is easy to observe that everything that happens in the 'future' (in the left part of the equation), can be described by only terms that happens in the 'present' (the right part), which are known or can be computed. This leads to the Cauchy problem formulation, which states that a first-order differential equation of the form:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0$$

has a unique solution if the initial conditions are given. Since the system's behavior at each time step depends only on known values from the previous step, the problem can be solved iteratively. The numerical integration approach ensures that if the initial condition is known, the solution at successive time steps can be computed progressively:

$$V_C(t_0 + \Delta t) = V_C(t_0) + V_D(t_0)$$

The same can be said about  $t_2$ , using  $V_C(t_1)$  to obtain  $V_C(t_2)$  and so on, repeating this step in an iterative way until the final time instant is obtained. Iterative operations are easy to implement in an algorithm with a computational cost in the order of  $O(NT)$ , where  $N$  are the number of nodes (molecules in the MolFCN simulation) and  $T$  is the number of time steps.

Let's applied what it was just mentioned in equation 2.4

$$V_C(t + \Delta t) = V_C(t) + \frac{\Delta t}{\tau} \cdot (V_D(t) - V_C(t)) \quad (2.6)$$

Equation 2.6 can be easily implemented in an iterative cycle in MATLAB, using a vector  $t$  which contains all the time steps. it is possible to insert it in a loop cycle where  $i$  is the index of the number of iterations performed (eq. 2.7).

$$V_C(t(i + 1)) = V_C(t(i)) + \frac{\Delta t}{\tau} \cdot (V_D(t(i)) - V_C(t(i))) \quad (2.7)$$

To ensure the effectiveness of this approximate solution, in picture 2.3 the exact solution and the approximated one are reported. The input considered for both solution is the negative time step reported in fig. 2.3.a. The visual representation reported in the last plot of the picture 2.3 shows both the exact and the approximated solution proposed in eq. 2.7, ensuring the effectiveness of the FEM method.

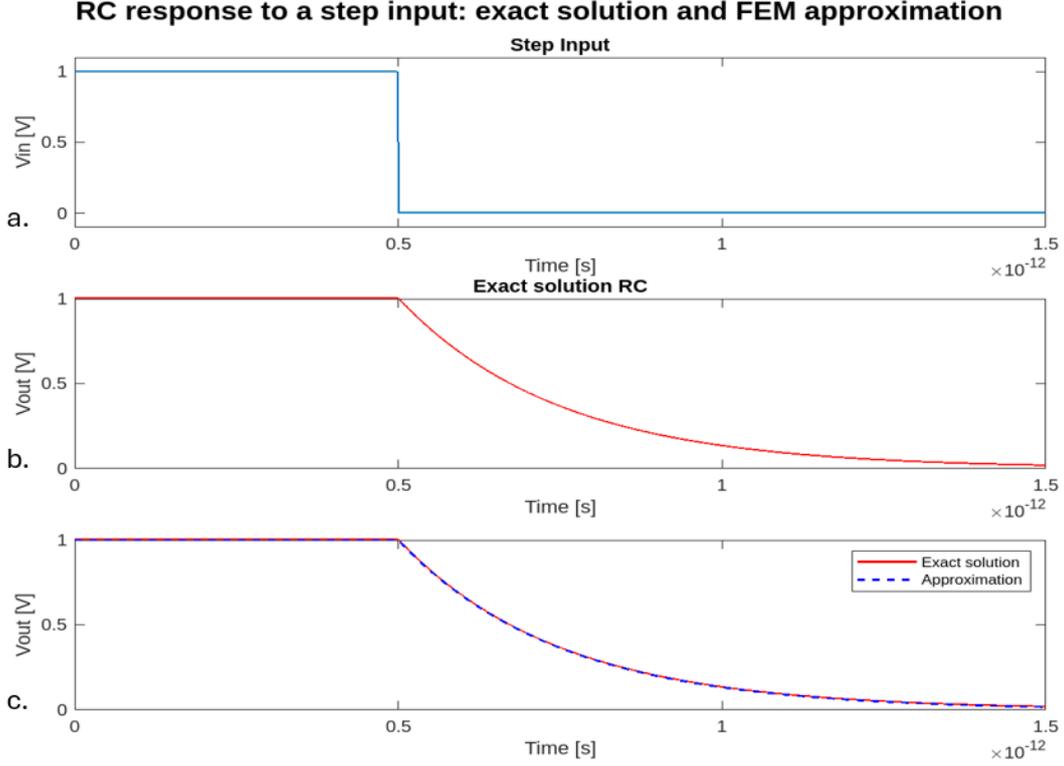


Figure 2.3: RC response to a step input: exact solution and FEM approximation, using the switch frequency of the molecule  $f_S = 640$  GHz

a. Negative step input at  $t = 0.5$ ps

b. Exact solution of the eq. 2.4 which is an exponential decay c. The approximated solution in eq. 2.6 using FEM method is reported in blue lines, while the exact solution in the continuous red line: from this visual representation it can be said that the approximated solution doesn't introduce any major discrepancy with the exact solution

## 2.2 Modeling

The electrostatic interaction between molecules is analyzed following the methodologies already implemented in SCERPA [4]. Molecules are considered as an aggregation of charges, generating an electric field in the surrounding space, which can be evaluated using electrostatic equations.

The voltage in a generic point in the space  $\mathbf{r}$  can be calculated using the fundamental electrostatic rule. The Euclidean distance between two point  $r_A$  and  $r_B$  is denoted as  $d(r_A, r_B)$  for clarity purpose. For simplicity, molecule are assumed to be in a vacuum, so the vacuum permittivity  $\epsilon_0 = 8.54 \cdot 10^{-12} C/Nm^2$  is used. The potential at  $\mathbf{r}$  due to a single charge  $Q$  at  $\mathbf{r}_Q$  is given by Coulomb's law:

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \cdot \frac{Q}{d(\mathbf{r}, \mathbf{r}_Q)} \quad (2.8)$$

Where  $\mathbf{r}_Q$  represents the position of the charge that generates the electric field. If multiple charge are present, the principle of superposition applies, meaning that the total electrostatic potential at  $\mathbf{r}$  is the sum of the potentials due to each individual charge:

$$V(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^N \frac{Q_i}{d(\mathbf{r}, \mathbf{r}_i)} \quad (2.9)$$

where  $N$  is the total number of charges in the system, and  $Q_i$  represents each charge located at position  $\mathbf{r}_i$ .

In a molecular wire or any electronic circuit implemented using MolFCN technology, multiple charges interact simultaneously, contributing to the overall electrostatic potential. Consequently, the electric field and the associated potential difference is the superposition of the fields produced by individual molecules. The voltage across each molecule can be evaluated using the electrostatic law, which states that the potential difference between two points is given by the circulation of the electric field:

$$V_{MUT} = \int_{\mathbf{r}_1}^{\mathbf{r}_2} \mathbf{E} \cdot d\mathbf{l} = V(\mathbf{r}_2) - V(\mathbf{r}_1) \quad (2.10)$$

In order to compute  $V(r_i)$  the superposition of eq. 2.9 coming from every charge in the molecule will be evaluated, according to eq.2.11.

$$V(r) = \frac{1}{4\pi\epsilon_0} \sum_{\beta=1}^{N_{AC}} \frac{Q_\beta}{d(r^\beta - r)} \quad (2.11)$$

where  $N_{AC}$  is the number of charges, put in position  $r^\beta$

This equation follows directly from Gauss's law and the conservative nature of the electrostatic field. Gauss's law states that the total electric flux through a closed surface is proportional to the enclosed charge:

$$\oint_S \mathbf{E} \cdot d\mathbf{A} = \frac{Q_{enc}}{\epsilon_0}$$

where  $d\mathbf{A}$  is an infinitesimal element of the surface  $S$ , and  $Q_{enc}$  is the total charge enclosed within  $S$ . Since the electrostatic field is conservative, its circulation over any closed path is zero:

$$\oint \mathbf{E} \cdot d\mathbf{l} = 0$$

This means that the electrostatic potential depends only on the initial and final positions, as shown in the equation 2.11.

Hence, the combination of equation 2.10 and eq. 2.11 are the ones used in this thesis for the determination of the differential potential across the Molecule Under Test (MUT) starting from its distribution of charges. This will be used for every molecule at every time step, as a result it is possible to evaluate the  $V_{MUT}$  of every molecule at every time step if its distribution of charges is known. This combine with the FEM formulation in eq.2.6 can gives as an evolution in time of the voltage across the molecule. The remaining computation that we need is how to find the distribution of charges related to a specific  $V_{MUT}$ .

## 2.3 The Voltage-Dependent Atomic Charge Transfer (VACT) function

The Voltage-Dependent Atomic Charge Transfer (VACT) function is a transcharacteristic function of a molecule that links a given voltage to its corresponding charge distribution. The VACT function is molecule-dependent and it is determined experimentally, which means that it is derived from a set of known data points. Once these points are available, interpolation can be used to estimate the charge distribution for voltages that lie between the known data points.

Interpolation theory is based on the assumption that the relationship between voltage and charge distribution follows a continuous, predictable trend. By using interpolation methods, such as MATLAB's 'interp1', an estimation of the values for the charge distribution at intermediate voltages can be build by constructing a function that passes through all of known points. The simplest interpolation method, linear interpolation, estimates values by connecting adjacent data points with straight lines, while more elaborated method like the spline interpolation use smooth curves to approximate the data more accurately.

$V_{\text{MUT}}$	$Q_1$	$Q_2$
1.9608	0.0796	0.9204
1.6946	0.0822	0.9178
1.4285	0.0848	0.9152
1.1624	0.0876	0.9124
0.8962	0.0909	0.9091
0.6301	0.1145	0.8855
0.4970	0.1849	0.8151
0.3640	0.2626	0.7374
0.2309	0.3406	0.6594
0.0978	0.4206	0.5794
-0.0352	0.4989	0.5011
-0.1683	0.5581	0.4419
-0.3014	0.6481	0.3519
-0.4344	0.7208	0.2792
-0.5675	0.8177	0.1823
-0.7006	0.8840	0.1160
-0.8336	0.9063	0.0937
-0.9667	0.8859	0.1141
-1.0998	0.9115	0.0885
-1.2328	0.9130	0.0870
-1.4990	0.9160	0.0840
-1.7651	0.9185	0.0815
-2.0313	0.9211	0.0789

Table 2.1: Tabella dei dati con  $V_{\text{MUT}}$ ,  $Q_1$  e  $Q_2$

From the data in tab 2.1 experimentally found for butane, the following VACT is obtained:

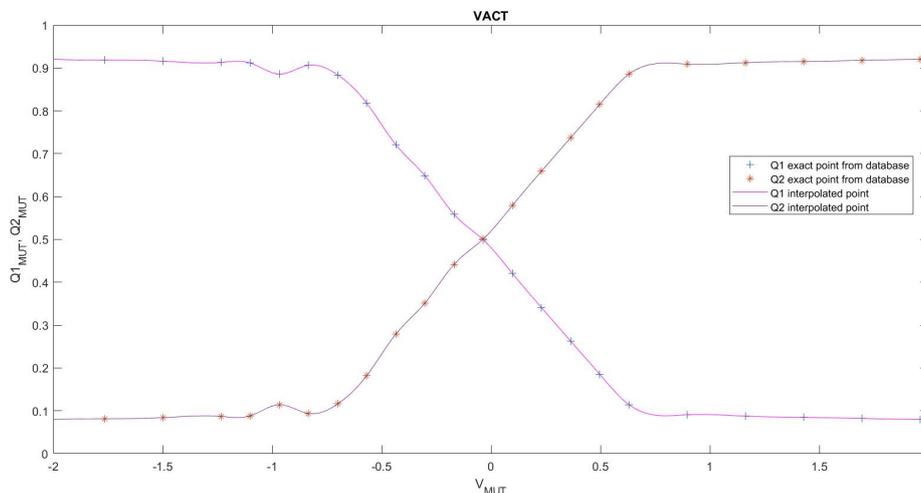


Figure 2.4: VACT of the butane molecule

The data point measured experimentally are highlighted, while the interpolation is shown like a continuous line.

As already anticipated, the bis-ferrocene molecule has 4 different center of aggregations. This is thanks to the bis-ferrocene geometry configuration, shaped like an Y: charge can be around the stick of the Y or in one of the 2 branches of the upper part of the Y, where the molecule can be polarized. The charge position along the vertical direction of the Y can be guided thanks to the use of an orthogonal field with respect to the direction of the information propagation, commonly denoted as the clock field.

The clock field can guide the charge and it can confined the electrons in the lower part of the Y, making the molecule stable in the null configuration, causing it to be "transparent" to what happens in the near regions. At the same time, it can also make the molecule more prone to be polarized if the opposite clock field is applied to the molecule, pushing the electrons in the upper region of the Y, enhancing the polarization of the molecule, hence the storage of information in its charge configuration.

The possibility to force the molecule in the upper or the lower region means also that for a different clock field we will have different VACT for the bis-ferrocene molecule. The figures below illustrate the variations in VACT for different typical values of the clock field. When a clock field is applied, electrons move in the opposite direction of it, since their charge is negative. Consequently, if the electric field along the z-direction is represented as in Fig. 2.6, electrons will be directed toward dot 3 and dot 4, leading to a nearly zero voltage output and a null charge configuration. In the absence of a clock field, the VACT of the molecule exhibits greater fluctuations due to a less linear relationship between charge distribution and the voltage across the molecule. Electrons are allowed to travel across the whole molecule, so part of the charge will be also stored in dot 3 and dot 4 without transmitting any information (fig.2.7). A positive clock field, on

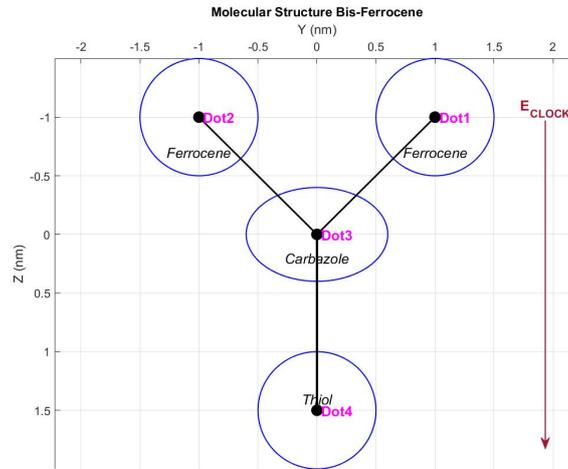


Figure 2.5: Here reported the bis-ferrocene molecule structure, with the direction of the electric field considered for the VACT

the other hand, promotes a more linear response, improving the reliability of information transmission. Molecules are confined in the two polarizing dot, so depending in the nearby molecule input, all the charge will flow in dot1 or in dot2 (fig. 2.8).

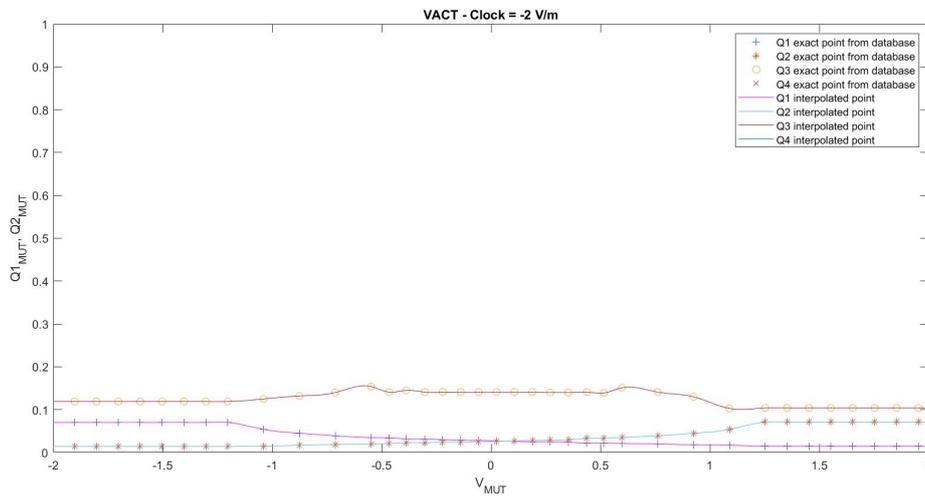


Figure 2.6: A.Clock = -2V/nm  
 If the clock is negative, molecules are forced to be stored in Q3 and Q4, so the voltage across the molecule is equal to zero.

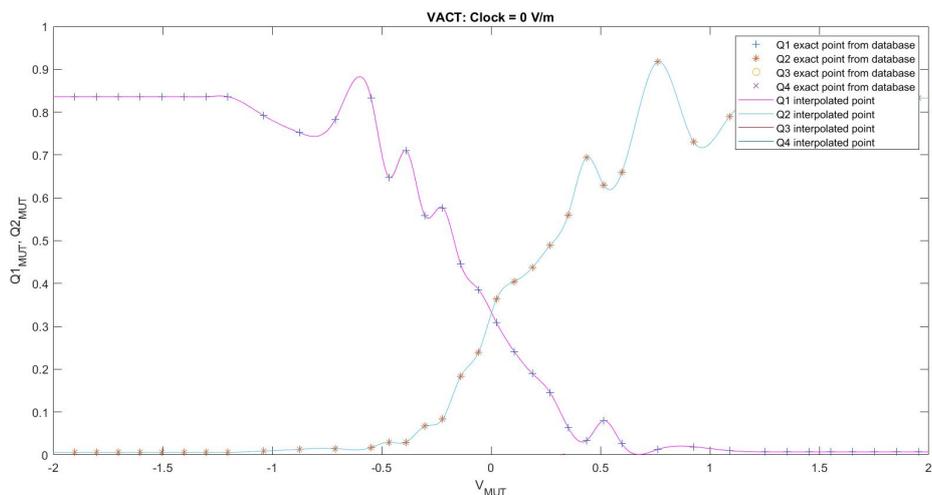


Figure 2.7: B.Clock = 0V/nm

If the clock is not present, the transmission can happen but we will have a more non linear behaviour, enhancing the probability of mistakes in the propagation.

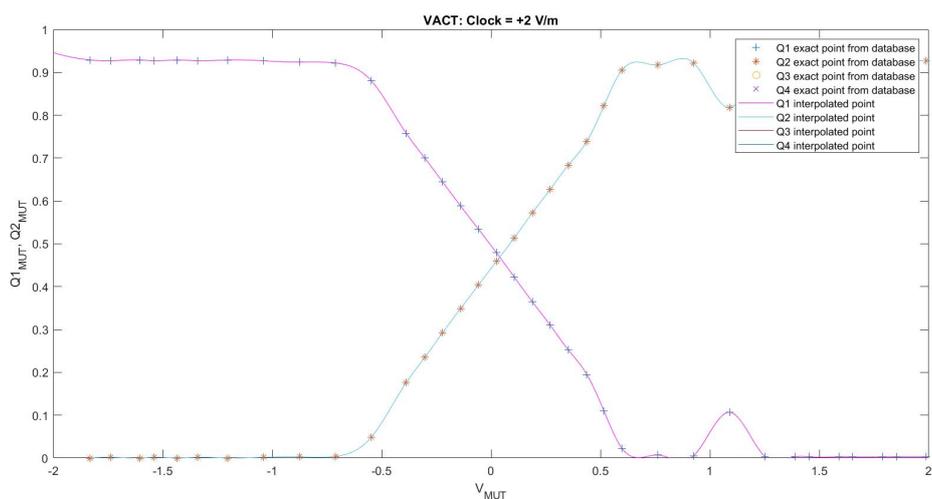


Figure 2.8: C.Clock = +2V/nm

With a positive clock, the VACT is almost linear.

## 2.4 Geometry of the circuits

Before going deep into the simulation, it is useful to understand how the molecules are positioned in the space.

In Figure 2.9, a molecular wire composed of four molecules is depicted. The driver molecule is positioned along the z-axis, centered at the origin. The two aggregated points are symmetrically placed with respect to the origin, using the width of butane,  $w = 0.6$

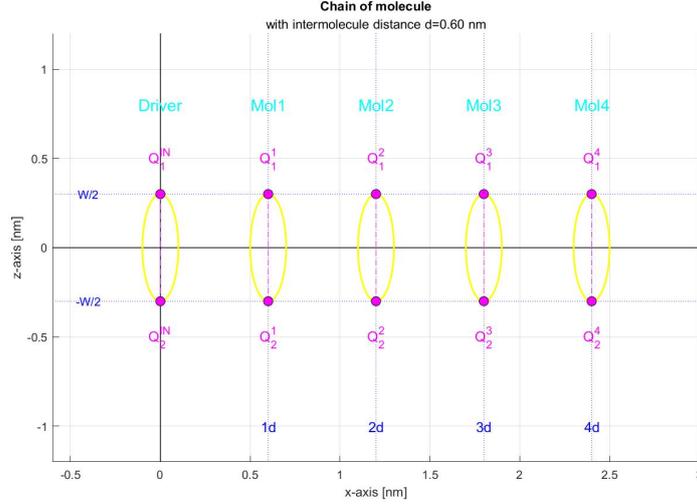


Figure 2.9: Wire of molecule

Here the wire is made by 4 molecule, each divided by one another by a distance  $d = 0.6\text{nm}$ . The width of the molecule is  $w = 0.6\text{nm}$

nm, as a reference. The centers of the molecules in the chain are placed at a distance  $d = w = 0.6\text{ nm}$  from each other. Each molecule is highlighted by a yellow circle, and its aggregated charge points are positioned along the  $z$ -axis:  $z = w/2$  for charge  $Q_1$  and  $z = -w/2$  for charge  $Q_2$ .

The intermolecular distance is one of the possible variables in the simulation. However, in this thesis, we have chosen to keep it equal to the molecular width to enhance the symmetry of the circuit. This is a common choice, as greater symmetry reduces the likelihood of errors. Moreover, this approach aligns with the design of majority voters (figure 2.10). The same principle is applied to the molecular wire, with a future applications in mind, where both circuits could be integrated.

The majority voter is a fundamental computational element in MolFCN. Unlike molecular wires, where signal propagation follows a more straightforward mechanism, the majority voter requires careful consideration of every input as a Quantum Cellular Automaton (QCA). Structurally, it consists of two molecules aligned as shown in Fig. 2.10.

In the majority voter, a molecular cell is influenced by three input molecules. Signal propagation follows the same fundamental principles as in molecular wires; however, an additional consideration is required: the effect of the charges from the other two inputs must be accounted for when determining the voltage across a molecule. The majority voter operates based on a "democratic" logic—meaning the output state aligns with the most dominant input. The input that is more prevalent will generate a stronger electric field, influencing the molecule to adopt the same polarization.

Once a cell has responded to the three inputs, an additional QCA element is introduced as a buffer for the output. This step is necessary because the first computational cell may retain a charge distribution that is not perfectly polarized. By adding a second QCA cell, the output is reinforced through positive feedback, ensuring more robust and

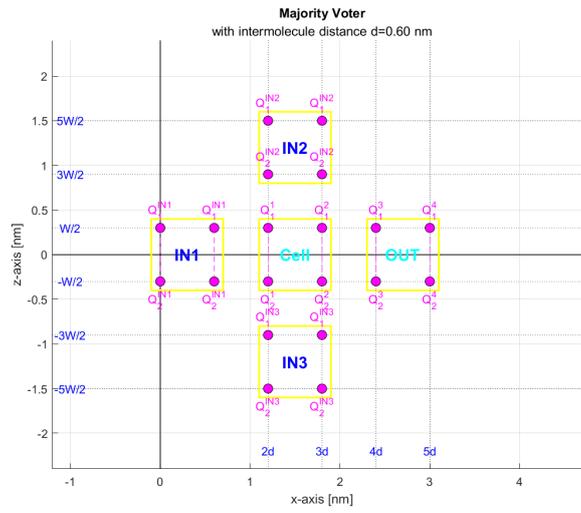


Figure 2.10: Space position of the molecule in a Majority Voter

reliable signal propagation.



## Chapter 3

# Charge dynamics Simulation algorithm

Three main algorithms are proposed: one for the simulation of a butane wire, one for the simulation of bis-ferrocene, and one for the simulation of the majority voter. The structure of the code allows the selection of the molecule used in the MOL-FCN circuit.

Every code is structured as follows:

1. Initialization of variables and definition of input data.
2. Calculation of the voltage induced by the input driver on the molecule in the wire using Eq. 2.11.
3. Execution of the function that computes the temporal evolution of charge dynamics.
4. Evaluation of Propagation Delay.
5. Plotting of the obtained result.
6. Analysis of the delay over the number of molecule, over the switching frequency and the intermolecular distance.

### 3.1 Initialization of variables and definition of input data

First, the temporal and geometrical parameters are defined, along with the number of molecules used in the wire. For the majority voter, as shown in Fig. 2.10, the circuit consists of two QCA subjected to three different inputs. Therefore, a wire of only four molecules is always considered. Additionally, in the majority voter circuit, the type of molecule must be specified.

Based on previous studies, the switching frequency of the molecules has been determined. In this work, we consider butane with  $f_S = 640$  GHz and  $f_S = 700$  GHz, while for bis-ferrocene,  $f_S = 250$  GHz and  $f_S = 300$  GHz are used.

Time is defined as a vector, with each time step taken in femtoseconds until a duration of several picoseconds is reached. The time step is chosen to be three orders of magnitude

smaller than the switching period, ensuring that it is sufficiently small to accurately capture the dynamic evolution of charge distribution within the molecules over time.

The geometrical plot showing the positions of the molecules in the circuit is also generated using the functions ‘**DrawChain**’ or ‘**DrawMaj**’, depending on whether a molecular chain or a majority voter is being simulated. Both functions require geometrical parameters such as intramolecular distance, intermolecular distance and the number of molecules. The output is the image shown in Fig. 2.9 or Fig. 2.10.

The initial voltage of all molecules in the chain is set to zero, ensuring that the charge distribution is perfectly balanced between the two polarizing dots. The charge is expressed as a percentage of the total number of charges in the molecule.

For the bis-ferrocene molecule, the non-polarizing aggregation points are assumed to be always null for simplicity. This is a reasonable approximation, as we expect that during information propagation, nearly all the charge will be distributed between the two polarizing dots,  $Q_1$  and  $Q_2$ .

The charge distribution of the input driver is also defined by assigning 90% of the charge to one dot and the remaining 10% to the other. For the majority voter, the two input values are stored in a vector in the form of 1s and 0s. The function ‘**DriverInput**’ is then used to map these binary values to the corresponding charge distributions at each QCA aggregation point. Finally, all input data is stored in the  $3 \times 4$  matrix ‘ $Q_{in}$ ’, where each row contains the charge distribution percentages for the four dots corresponding to each input.

### 3.2 Calculation of the voltage induced by the input driver on the molecule in the wire

Before implementing the time-dependent algorithm, the voltage induced on each molecule by the input driver is calculated by applying Eq. 2.11 in combination with Eq. 2.10.

For the molecular wire, only two accumulated charges are considered in the summation. Therefore, Eq. 2.11 and Eq. 2.10 are computed simultaneously. The result is a vector,  $vin_D$ , which contains the voltage induced by the input at each molecule.

For the majority voter, the same calculation of  $vin_D$  is performed. However, instead of having a single molecule as a driver, the majority voter consists of three QCA, each composed of four dots, leading to a total of 12 accumulated charges. Consequently, for each input, the voltage induced by each charge in the QCA is computed and summed into an accumulative variable.

### 3.3 Execution of the function that computes the temporal evolution of charge dynamics

Now everything is ready for the core of the algorithm, which consists in the computation of eq. 2.7 repeated for every time step for every molecule. The function ‘**Evolution in time**’ is used in all the simulations, since once the voltage induced by the input is known, the majority voters can be considered a molecular wire made of 4 molecules.

All the input and the result of the calculation made so far are put as input of this function. Initially, the VACT is extrapolated from the known data of the molecule used. This also is done recalling the function VACT or VACT\_bis.

Then the actual time loop is performed, calculating for every molecule the distances between the dots in the chain. In  $r11[i,j]$  all the distances between Q1(i) and Q1(j) or Q2(i) and Q2(j) are calculated, while in  $r12[i,j]$  all the distances between Q1(i) and Q2(j) are calculated. Then for every time instant, for every molecule the sum of all the contribution of the voltage induced on dot1 and dot2 is made for the MUT, calculating the input on MUT coming from all the other molecule in the circuit. At the calculated  $vin$ , the voltage coming from the driver is added, then the eq. 2.7 is performed, finding the  $V_{MUT}$  of the next time instant. Hence the VACT calculated previously is used to derive the relative distribution of charges of the MUT at the following time input, which will be used in the next time step repeating the same step again with new input (see pseudocode in Algorithm 1).

### 3.4 Evaluation of Propagation Delay

This function calculates the delay based on when the last molecule reaches convergence on its final distributions Q1 and Q2. Convergence is monitored for both Q1 and Q2, ensuring that for a number of consecutive time steps, the condition  $|Q1(t) - Q2(t)| < 0.005$  holds. In this thesis, the number of times that the convergence condition needs to be verified is 10, while the convergence threshold is set at 0.5% of the total charge in the molecule. Furthermore, a check is performed on the value of Q1 and Q2: the delay is only considered if Q1(t) differs from its final value by at most the convergence limit. This additional check ensures that the delay corresponds to actual propagation rather than being influenced by the initial stable state of the molecule before information transmission occurs.

When the algorithm detects convergence, the corresponding time index is recorded. The output delay is then calculated as the convergence index minus the number of consecutive steps required for the molecule to be considered stable (algorithm 2).

### 3.5 Plotting of the obtained result

At the end of each of the three main MATLAB scripts, plots are generated to visually represent the results.

Additionally, for the majority voter, the output is assigned to one of the possible configurations: 0 or 1. If the output cannot be reliably associated with either, it is assigned a value of 9. This approach provides an immediate indication of the propagation status, allowing for quick identification of the final result of the computation performed.

---

**Algorithm 1** Pseudocode of the EvolutionInTime function

---

```

1: function EVOLUTIONINTIME( $t, V_{\text{tot},D}, Q_1, Q_2, v_{\text{Mol},\text{in\_time}}, \text{molecule}, n, d, w, \tau, \dots$ )
2:   Initialize the molecular chain layout ▷ Init stage
3:    $k = 1/4\pi\epsilon_0$ 
4:   Compute distance matrices  $r_{11}$  and  $r_{12}$ 
5:   if molecule == 'butane' then ▷ Load VACT for butane
6:     Load VACT data:
7:   else if molecule == 'bis-ferrocene' then ▷ Load VACT for bis-ferrocene
8:     Load VACT data for three clock field conditions:
9:   end if
10:  for each time step  $i$  do
11:    for each molecule  $j$  do
12:      Exclude molecule  $j$  from its own interaction calculations
13:      Compute electrostatic contributions from neighboring molecules:
14:         $q1r1 =$  effect of all the  $Q_1$  of the molecule on dot 1 of molecule  $j$ 
15:         $q1r2 =$  effect of all the  $Q_1$  of the molecule on dot 2 of molecule  $j$ 
16:         $q2r1 =$  effect of all the  $Q_2$  of the molecule on dot 1 of molecule  $j$ 
17:         $q2r2 =$  effect of all the  $Q_2$  of the molecule on dot 2 of molecule  $j$ 
18:      Compute net potential at molecule  $j$ :
19:         $V_{\text{in},j} = V_{\text{tot},D,j} + k \cdot (q1r1 + q2r1 - q1r2 - q2r2)$ 
20:      Update molecular voltage:
21:         $v_{\text{Mol},j}(i+1) = v_{\text{Mol},j}(i) + \Delta t \cdot (V_{\text{in},j} - v_{\text{Mol},j}(i))$ 
22:      if molecule == 'butane' then
23:        for each VACT entry  $p$  do
24:          if  $|v_{\text{Mol},j}(i+1) \cdot e * V_{\text{VACT}}(p)| < \Delta V$  then
25:            Update  $Q_1(j, i+1)$  and  $Q_2(j, i+1)$ 
26:          end if
27:        end for
28:      else if molecule == 'bis-ferrocene' then
29:        for each VACT entry  $p$  do
30:          if  $|v_{\text{Mol},j}(i+1) - e * V_{\text{VACT}}(p)$  of relative clock  $| < \Delta V$  then
31:            Update  $Q_1(j, i+1)$  and  $Q_2(j, i+1)$ 
32:          end if
33:        end for
34:      end if
35:    end for
36:  end for
37:  if molecule == 'butane' then
38:    Return  $Q_1, Q_2, v_{\text{Mol},\text{in\_time}} \cdot e$ 
39:  else if molecule == 'bis-ferrocene' then
40:    Return  $Q_1, Q_2, Q_3, Q_4, v_{\text{Mol},\text{in\_time}} \cdot e$ 
41:  end if
42: end function

```

---

---

**Algorithm 2** Pseudocode of the proposed algorithm for finding delay

---

```
1: function FINDDELAY( $t, Q_1, Q_2$ )
2:   Initialize:
3:     convergence_reached = 0
4:     flag_convergence = 0
5:     convergence = 0.005
6:     c = 1
7:     delay = 99
8:     num_stable = 10
9:     while convergence_reached < num_stable AND c <= length(t) do
10:      Increment c
11:      if  $|Q_1(c) - Q_1(c-1)| < \text{convergence}$  AND  $|Q_2(c) - Q_2(c-1)| < \text{convergence}$ 
AND  $|Q_1(c) - Q_1(\text{last})| < \text{convergence}$  AND  $|Q_2(c) - Q_2(\text{last})| < \text{convergence}$  then
12:        if flag_convergence == 1 then
13:          Increment convergence_reached by 1
14:        end if
15:        Set flag_convergence = 1
16:      else
17:        Set flag_convergence = 0
18:        Set convergence_reached = 0
19:      end if
20:      if convergence_reached == num_stable then
21:        Set delay =  $t(c - \text{num\_stable})$ 
22:        Break
23:      end if
24:    end while
25:    Output delay
26: end function
```

---

### 3.6 Analysis of the delay over the number of molecule, over the switching frequency and the intermolecular distance.

The last script analyzes the delay of a molecular wire system (specifically for butane molecules) based on various parameters such as the number of molecules ( $n$ ), molecular spacing ( $d$ ), and switching frequency ( $f_{\text{switch}}$ ).

- Delay vs. Number of Molecules:  
It calculates the delay for different numbers of molecules in the wire (from 3 to 30), keeping the distance and switching frequency constant. The result is plotted with delay on the y-axis and the number of molecules on the x-axis.
- Delay vs. Distance:  
The delay is calculated for varying molecular distances (from 0.3 nm to 0.63 nm), keeping the number of molecules and switching frequency fixed. The delay is plotted against the distance.
- Delay vs. Switching Frequency:  
The delay is also calculated for different switching frequencies (from 450 GHz to 900 GHz), keeping the number of molecules and distance constant. The delay is plotted against the switching frequency.

Each analysis section uses the ‘simulationButane’ function, which is the script used for the simulation previously described turned into a function to compute the delay for the respective parameters. Then the results are plotted for visual interpretation.

# Chapter 4

## Result of the Simulation

The previously described algorithm enables simulations to determine the delay in MolFCN circuits. In this chapter, the plots generated from previously described simulations are presented, analyzing molecular wires and majority voters made of both butane and bis-ferrocene molecules. Specifically, the butane molecular wire is examined under various parameterizations. The delay is also investigated as a function of molecular spacing, switching frequency and the number of molecules in the chain, with the objective of studying its dependence from this parameters.

### 4.1 Wire of molecule

Here we will analyzed what is the outcome of the molecular wire simulation.

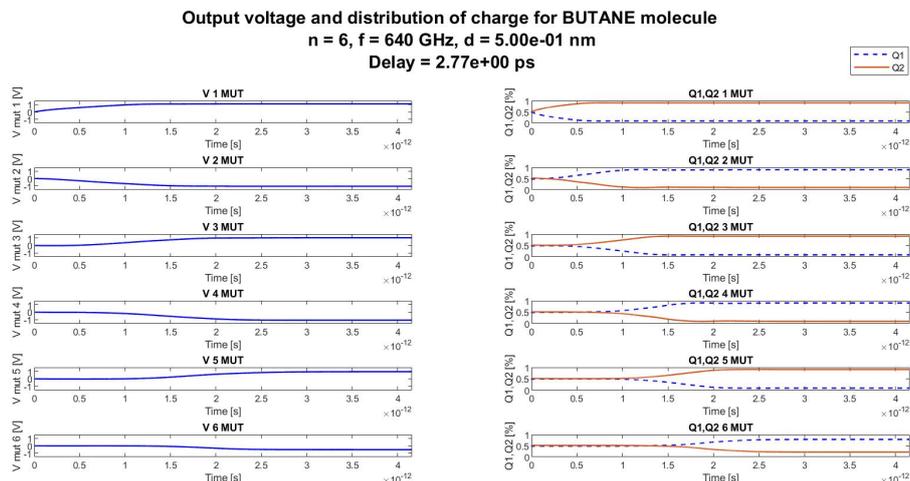


Figure 4.1: Evolution in time of the molecule in the molecular wire: On the right the evolution of the voltage across the molecule is represented, while on the left part the focus is made on the evolution of the distribution of charges

The previous figure 4.1 represents the time evolution of what happens along the butane molecular wire. Every molecule is represented with both the evolution in time of the molecule and its relative distribution of charges. Also, in fig. 4.2 and fig. 4.3 it is possible to focus on what happens in the first and in the last molecule of the wire

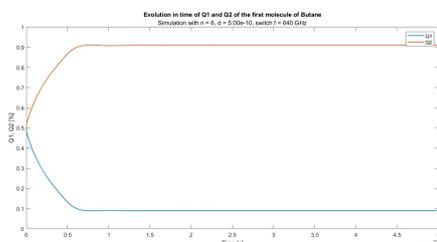


Figure 4.2: First Butane molecule of the molecular wire with a frequency switch = 640GHz

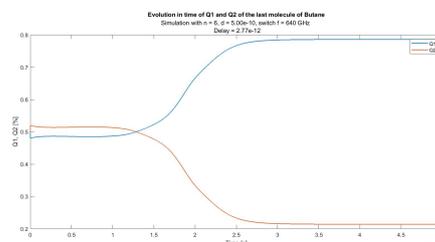


Figure 4.3: Last Butane molecule of the molecular wire with a frequency switch = 640GHz

It is possible to observe the behaviour of every molecule and determine when the input arrives. In fact, the first molecule (fig 4.2) reacts instantaneously to the input, since it is the nearest one. The switching frequency taken as a reference is equal to 640 GHz, so the relative  $\tau \cong 0.25$  ps. The first molecule reaches the stabilization after about 0.6 ps, which is consistent with the chosen  $\tau$ . In fact, from the analysis of exponential transfer functions, it is known that the response time to reach 90% of the final value is given by  $\ln(10)\tau \cong 0.575$  ps. The information arrives in the last molecule with a delay of 2.45 ps (fig 4.3), hence the total delay of the wire is less than the delay of 6 molecule considered separately. This was expected, in fact every molecule in the chain is subjected from  $t = t_0$  to the electric field coming from the driver, the molecule will start to react before the real arrival of the information on the previous molecule, enhancing the speed of the information in the last part of the wire.

The same graphic are reported below for the bis-ferrocene molecule (fig 4.4).

Analogue considerations can be made for the molecular wire with the bisferrocene molecule, at least for what concern the behaviour obtained with a positive clock. The switching frequency of the molecule is 250 GHz, so the associate  $\tau \cong 0.63$  ps, which imply that the bis-ferrocene molecule has a slower switch frequency than the butane one. In this case, the overall delay is of 4.72 ps, due to the slower frequency switch of the bisferrocene molecule. The present on the spike in the first molecule is a possible direct consequence of the spike which was also present on the VACT of the bisferrocene with a positive clock (fig. 2.8).

Regarding the influence of the clock on the delay of the molecular wire, fig 4.7 shows the resulting distribution of charges when there is a negative clock field applied. Both Q1 and Q2 are equal to zero, since the whole charges are confined in the other 2 dots of the molecule, hence no information can be propagated.

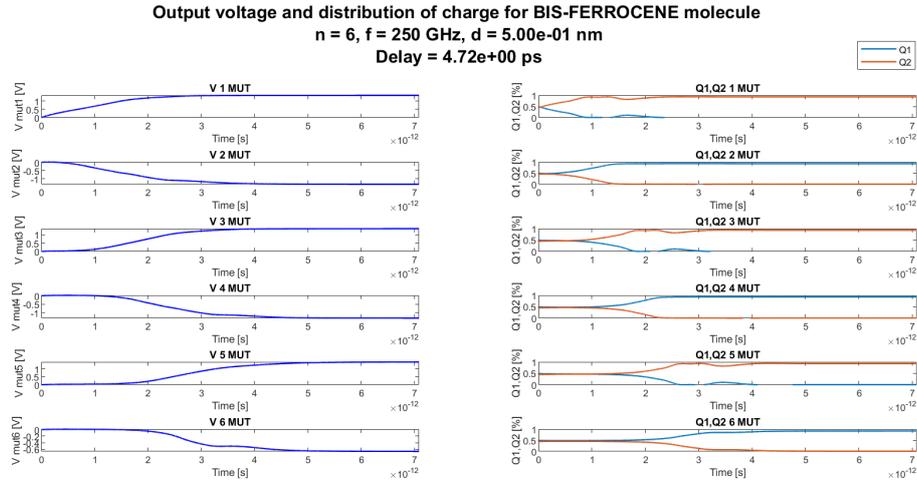


Figure 4.4: Evolution in time of the molecule in the molecular wire: On the right the evolution of the voltage across the molecule is represented, while on the left part the focus is made on the evolution of the distribution of charges

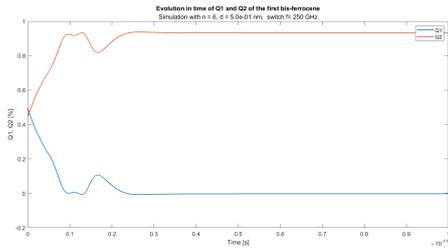


Figure 4.5: First bisferrocene molecule of the molecular wire with a frequency switch = 250GHz

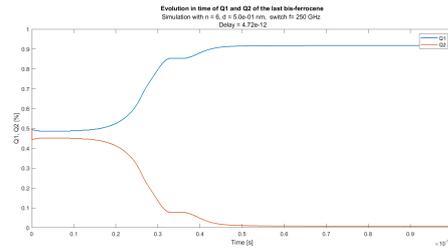


Figure 4.6: Last bis-ferrocene molecule of the molecular wire with a frequency switch = 250GHz

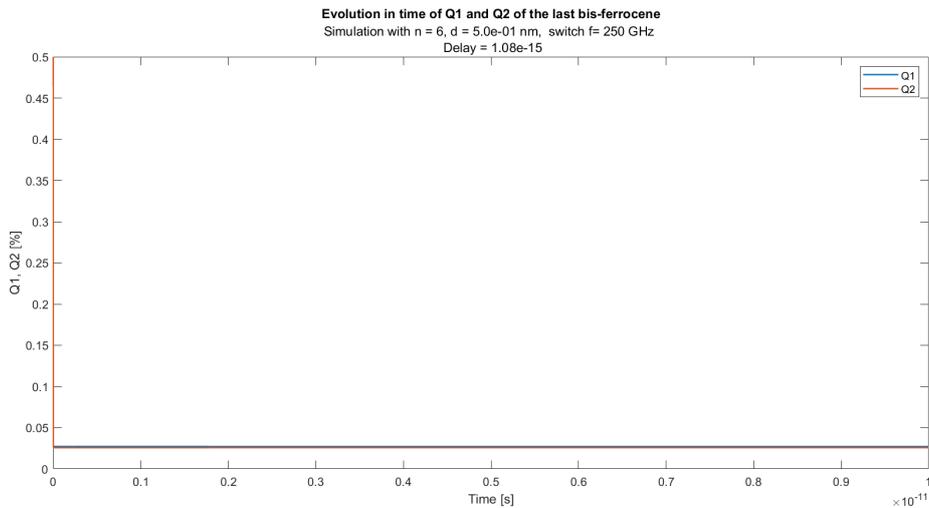


Figure 4.7: Last molecule of the molecular wire with negative clock: charge distributions of Q1 and Q2 are fixed to zero since the whole distribution is in the other dot. Hence, the molecule is in a null configuration

Without the clock, the transmission of information is allowed, but some of the charges are lost in the Q3 and Q4 (even if fig. 4.8 shows us only Q1 and Q2). There are more fluctuations in the distribution of charges, as a consequence of a non-linearity of the VACT in the absence of the clock field (fig. 2.7). This leads also to a major delay, which in this case is of 5.16 ps, while the delay for the same molecular wire with the positive clock was of 4.72 ps.

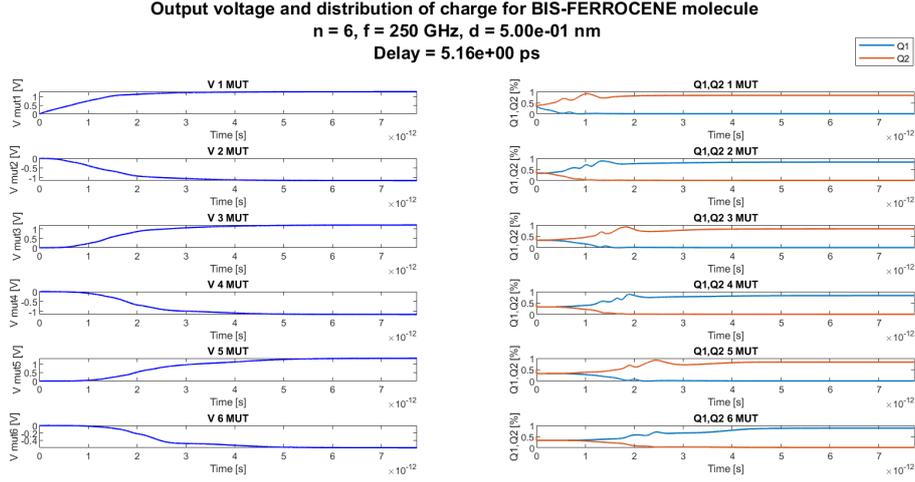


Figure 4.8: Molecular wire without clock

## 4.2 Majority Voter

In the majority voter the molecules are organized in QCA, like is showed in fig. 2.10. For a better understanding of the results of the simulation, the structure of the last QCA which will be the output of the circuit, made of 2 molecules is reported in fig. 4.9.

It is possible now to analyze what happens on the aggregation of charges of the output cell.

The fig. 4.10 reports what happens in a majority voter composed of butane molecule. The worst scenario is taken, so the 3 input are not all equals. The outcome is reported in fig. 4.11.

As highlighted in the graphic, it is observed that the output the final molecule reaches a different charge distributions. The penultimate molecule has a very defined state, where the Q1 held the 0.9 of the charges and Q2 has the remaining 0.1, while the last molecule has a less defined concentration of 0.83 of charges in Q2 and the 0.16 in Q1. This can be explained by the fact that the last molecule does not have another cell after it that can enforce its distribution of charges, which helps maintain a more defined charge state. The delay of the circuit is of 1.3 ps which is less than the molecular wire made of 6 molecules, since there are only 4 molecule in the majority voter.

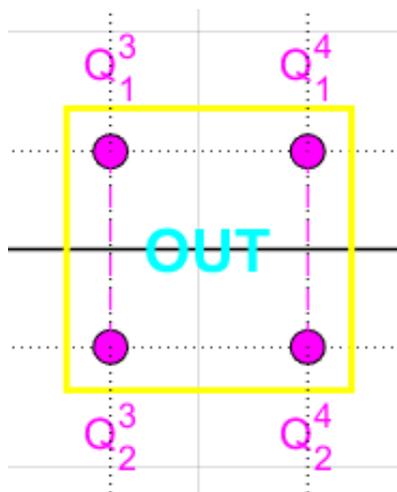


Figure 4.9: Last 2 molecules of the Majority Voter, forming an unique QCA

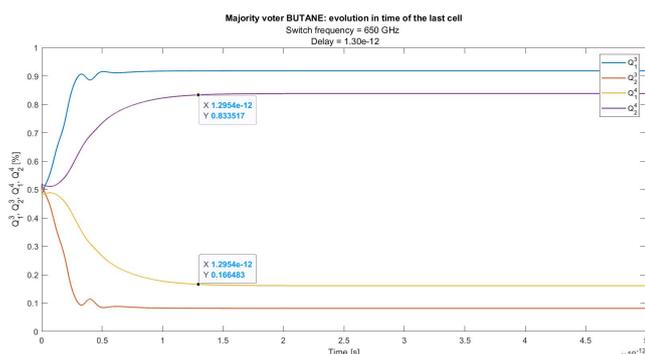


Figure 4.10: A

```

delay =
    1.2954e-12

in =
    1    0    1

last =
    1
    
```

Figure 4.11: B

Figure 4.12: A. Last molecule of the Majority Voter with Butane molecule, calculated with a  $f_s = 650$  GHz, with a resulting delay = 1.2954 ps  $\cong$  1.3 ps, while on B we can observe the input, the output associated to the configuration shows in A and the exact delay

Analogue considerations can be made for the majority voter composed of bisferrocene molecule in presence of the positive clock electric field (fig. 4.13 and outcome in fig.4.14). The delay in this case was of  $d = 3.98$  ps, which is much more than the circuit made of butane.

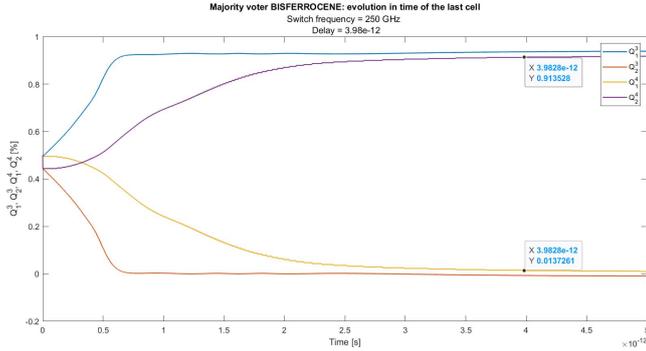


Figure 4.13: A

```

delay =
    3.9828e-12

in =
    1     0     1

last =
    1
    
```

Figure 4.14: B

Figure 4.15: A. Last molecule of the Majority Voter with Bis-Ferrocene molecule, calculated with a  $f_S = 250$  GHz, performed in the case where 2 input are the ones showed in B, along with the final Output (denoted as last) and the exact delay.

### 4.3 Study of the delay as a function of n, d and the frequency switch

Multiple simulations are performed, in order to allow the study of the delay over multiple parameters. In this thesis the delay was analyzed varying one variable at a time, fixing the others at a reference value 4.1.

Parameter	Value
Switching Frequency ( $f_{\text{switch}}$ )	640 GHz
Number of Molecules ((n))	8
Distance ( $d$ )	0.5 nm

Table 4.1: Reference values for simulation parameters

In plot fig. 4.16 it can be observed that, as expected, the delay is greater if the switching frequency decrease, due a faster stabilization of charge distributions across the molecular wire.

The outcome of the delay as a function of N (fig. 4.17). From this analysis, it is peculiar the fact that the outcome has multiple spike, in fact even if the general trend is to increase the delay increasing the number of molecule, it seems that chain with odd molecules have a faster transmission than the even one. This is a direct consequence of the slight asymmetric behaviour of the butane, to demonstrate it in fig. 4.18 the simulations are run with different input. The general trend is the same, but in fig. 4.17 the faster chain are characterized by an even number of molecule, while, as already stated previously, in fig. 4.17 the transmission were faster with the odd chain.

The delay as a function of the distance between molecule in fig. 4.19 reports different

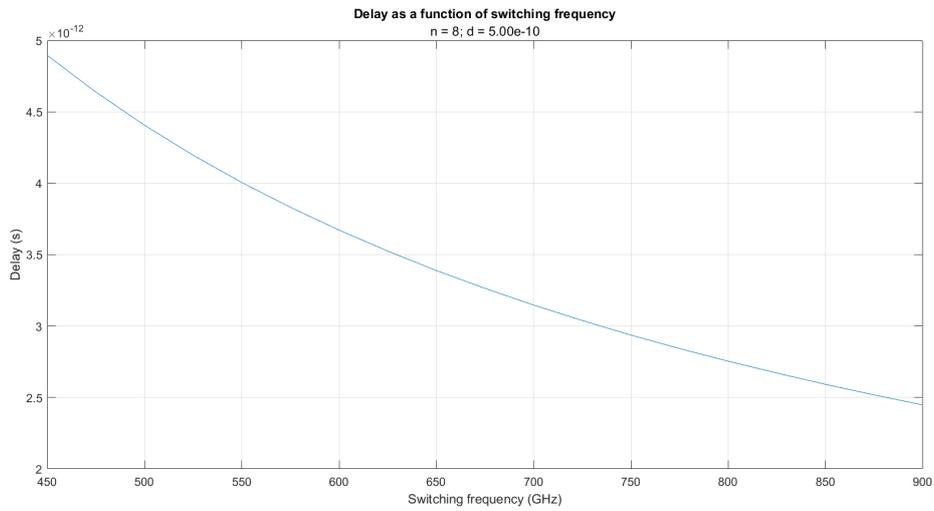


Figure 4.16: Delay as a function of the switch frequency of the molecule

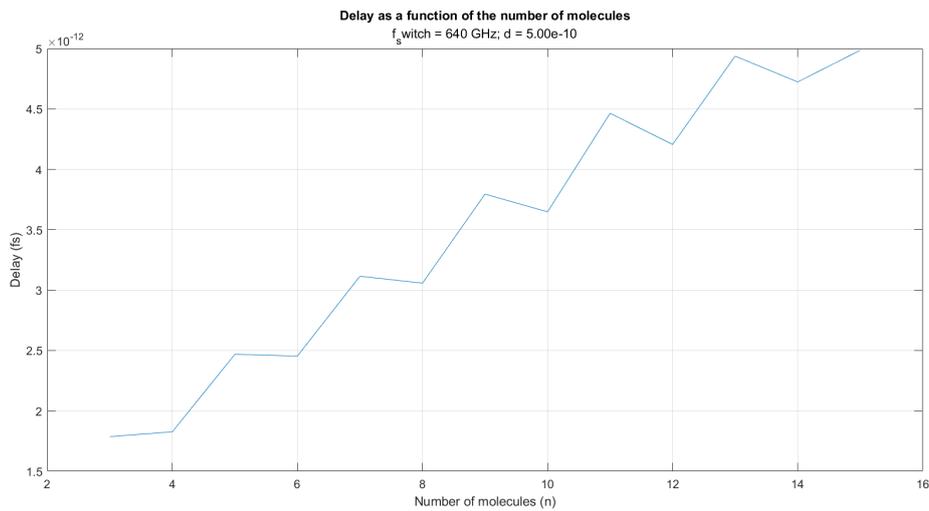


Figure 4.17: Delay as a function of the number of molecule that composed the molecular wire

behavior that can be divided in 3 regions:

1.  $d$  less than 3.75 nm
2.  $d$  between 3.75 nm and  $d = 4.75$  nm
3.  $d$  over 4.75 nm

In the central region, the relationship between delay and intermolecular distance can be considered linear, exhibiting direct proportionality between  $d$  and the delay. Whereas

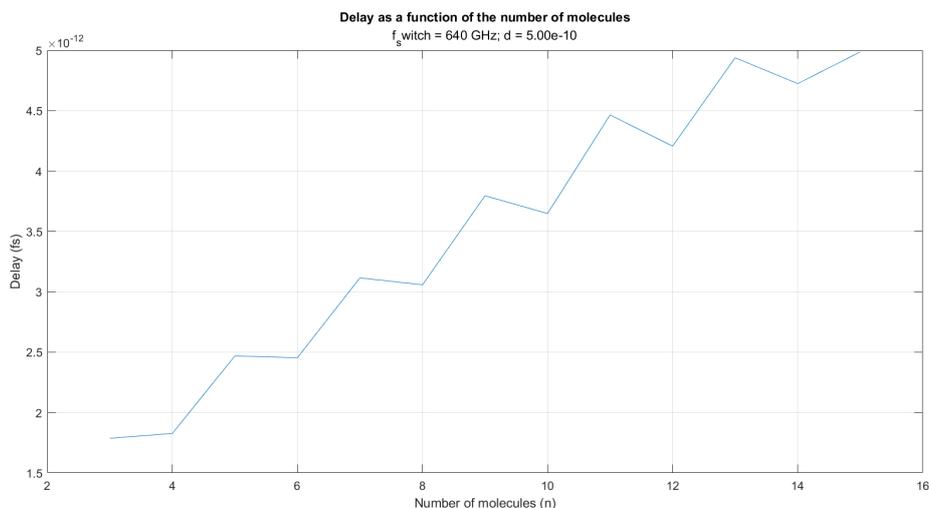


Figure 4.18: Delay as a function of the number of molecule with the opposite input

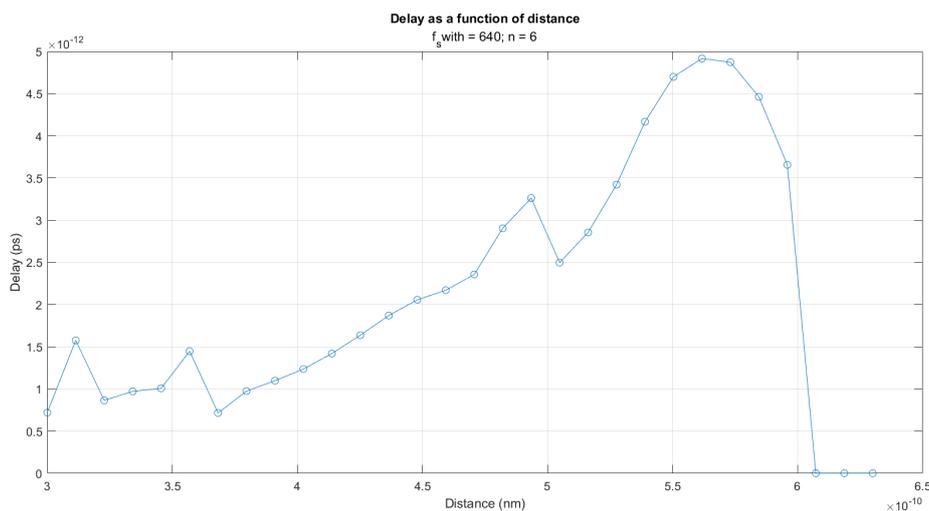


Figure 4.19: Delay as a function of the intermolecular distance

where  $d$  is too low, molecule are too close to each other, so they won't allow the molecule to transmit the information because they will have too influences coming from the near molecule. This will have an unstable outcome, for example we take figure 4.20, where we have the distribution of charges of the last molecule of a molecular wire of butane where  $d = 0.32$  nm. The delay obtained can't be considered, since the charges never reaches a real stabilization. When the molecules are too far apart, the influence of one molecule on the next is significantly reduced, causing the information to be disrupted and eventually lost along the chain. The delay is calculated when convergence is reached, without verifying whether the information has been transmitted correctly. As a result, if the information is lost before reaching the last molecule, the charge will appear to

converge almost instantaneously to the final value within the considered time frame. However, this occurs only because it remains nearly identical to the null configuration, without proper alignment to the input, like fig. 4.21 highlights.

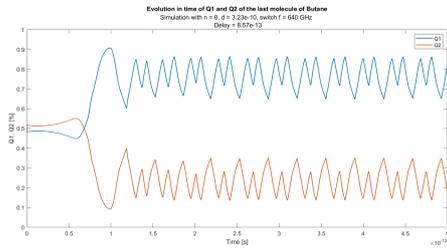


Figure 4.20: Charge distribution when  $d$  is too low.

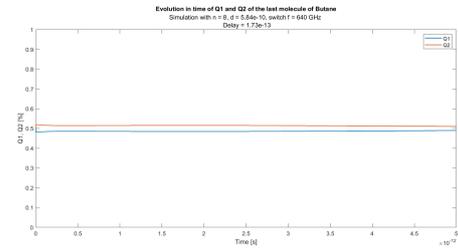


Figure 4.21: Charge distribution when  $d$  is too high.



## Chapter 5

# Conclusion and future perspectives

This study has explored the dynamic simulation of electronic devices based on MolFCN molecular technology, taking a further steps toward the knowledge of this technology. The analysis of the dynamics of the molecules during the propagation of information can add a better understanding of what happens in the distribution of charges of the molecule and the simulations carried out in MATLAB allowed for the evaluation of the molecular dynamics can be a starting point for it.

The result obtained from the study of the delay of the butane molecule, has contribute to widen and deepening our understanding of the butane molecule. The more interested result was the dependency of signal transmission delay on molecular distance. In a specific range, the delay was found to be approximately linear with distance, indicating direct proportionality. However, when molecules were placed too close together, strong interactions led to unwanted effects, such as excessive coupling, which hindered proper signal transmission. On the other hand, when molecules were too far apart, their influence on one another diminished, leading to information loss along the chain. This highlights the importance of optimizing molecular spacing to ensure reliable signal transmission while minimizing delay and gives the basic guidelines for the construction of MolFCN circuits.

Even though not further analyzed in this work, the fundamental elements necessary for studying the dependencies of the bis-ferrocene molecular wire on molecular wire parameters have been established, with the possibility of also focusing on the influence of the clock on the circuit.

Despite these promising findings, several challenges remain in the development of FCN-based molecular electronics. One of the key limitations is the complexity of molecular interactions, which require highly accurate models to predict their behavior under different conditions. Environmental factors, such as temperature fluctuations and quantum effects, also play a crucial role in determining the performance of these devices, further complicating their practical implementation, which will be the focus on future researches.

In conclusion, this thesis has provided a model that can be used for the studies of a comprehensive analysis of the dynamic behavior of FCN molecular devices, offering

valuable insights into their potential applications in nanoelectronics. By exploring the interplay between molecular structure, signal propagation, and external influences, this work lays the foundation for future advancements in molecular-scale computing.

The results presented here reinforce the idea that molecular electronics can serve as a viable alternative to traditional semiconductor-based technology in specific applications, particularly where ultra-miniaturization and low-power operation are required. While challenges remain, continued research in this field, supported by both theoretical and experimental efforts, will be key to unlocking the full potential of molecular computing and advancing towards a new era of nanoscale information processing.

# Appendix A

## MATLAB code

### A.1 simulationButane.m

Below is the MATLAB code used for simulating the dynamic behavior of a chain of butane molecules subjected to time-varying electric fields. This simulation aims to analyze the behavior of the molecule, including its delay in switching behavior.

```
% clc;
% clear all;
% close all;

% Uncomment the comment below to create a function that can be called
% multiple times for analysis purpose (Also the end at the end of the
% script)

% function delay= simulationButane(n,d,f_answer)

% n = number of molecule in the chain
% f_answer = experimental value of the switch frequency of the molecule
% d = intermolecular distance
n = 50; % comment if the function is used
f_answer = 650 * 109; % Hz comment if the function is used
d = 0.5*(10-9); % meter

% DYNAMIC SIMULATION OF A CHAIN OF BUTANE MOLECULE
% The aim of the code is to create a simulation of a chain of BUTANE
% molecule, finding its dynamic response in order to analyze its behavior
% in time and the relative delay
% In the simulation we use a polarized molecule as a driver input to the
% chain and we assumed to have perfectly balanced molecule at t=0

molecule = 'butane';
% t = time vector
```

```

t = 0:10(-15):10*10(-12); % every time step is 1fs
timeF = length(t); % Final time instant

% tau = time constant of the molecule (equivalent RC constant)
tau = 1/(2*pi*f_answer); % s

% w = length of the molecule = distance between dot1 and dot2 of
%the molecule
w = 0.45*(10(-9)); % meter -> CHARACTERISTIC OF THE MOLECULE

% COMMENT to hide a visual representation of the molecule position in
%space
DrawChain(w,d,n)

% Electric constants
e = 1.6 * 10(-19); % Coulomb -> charge of the electron
eps0 = 8.854187817* 10(-12); % Faraday/m = electric permittivity in air
k = 1/(4*pi*eps0); % electrostatic constant

% Position of the charge of the external driver molecule
r1_D = [0; w/2];
r2_D = [0; -w/2];

% Distribution of charge in the driver = polarized molecule
Q1_D = 0.9; % C
Q2_D = 0.1; % C

% Q1/2_0 = initial value of the charge of the molecule in the chain
Q1_0 = 0.5 * ones(n,1); % C
Q2_0 = 0.5 * ones(n,1); % C

% Initial voltage of the molecule
vmol_0 = zeros(n,1); % since Q1_0 = Q2_0 = 0.5 * e

% Q1/2 = matrix n X timeF -> n = number of molecule in the chain;
%time instant so in i-row there is the evolution in time of Q1/2 of the
%i-molecule in i-column we have the photo of how the Q1/2 are at time i
Q1 = zeros(n,timeF);
Q2 = zeros(n,timeF);
Q1(:,1) = Q1_0;
Q2(:,1) = Q2_0;

% vMol_in_time = vector with vmol at every time instant, with initial
%value vmol vMol_in_time = zeros(n,length(t));

```

---

```

vMol_in_time(:,1) = vmol_0;

% x position of every molecule (same for Q1 and Q2)
% y position of Q1 -> always equal to w/2
% y position of Q2 -> always equal to -w/2
x = d : d : n * d;

% Calculation of vin (constant for every molecule since we impose that
%Q1_D and Q2_D are constant)
% vin = vector [n] where vin(i) = vin_D for molecule i
vin_D = zeros(n,1);
vin_D = k * (Q1_D ./ x + Q2_D ./ sqrt(x.^2 + w^2) - ...
            ...Q1_D ./ sqrt(x.^2 + w^2) - Q2_D ./ x);

% Function with the computation for every t(i)
[Q1, Q2, vMol_in_time] = EvolutionInTime(t, vin_D, Q1, Q2, ..
            ..vMol_in_time, molecule, n, d, w, tau);

% xt = new point for time (should be different and less than before
t_T = 0: 6*10-16 : t(end);
% Extracting the linear polynomial for the previous point, then it will
%use
% the linear polynomial to find the relative vmol for the new xt
vmol_T = interp1(t, vMol_in_time(n,:), t_T);
Q1_T = interp1(t, Q1(n,:), t_T);
Q2_T = interp1(t, Q2(n,:), t_T);
% Find delay of the convergence in the last molecule
delay = FindDelay(t_T, Q1_T(:), Q2_T(:));

%% plotting -> uncomment the plot that you need
% Uncomment for the plot of Q1 and Q2 of the last molecule of the chain
%only
figure;
plot(t_T, Q1_T, t_T, Q2_T, 'linewidth', 1.2);
xlabel('Time [s]')
ylabel('Q1, Q2 [%]')
title('Evolution in time of Q1 and Q2 of the last molecule of Butane',...
        ..'FontWeight', 'bold')
subtitle(sprintf('Simulation with n = %d, d = %.2d, switch f = %d GHz \n..
        ..Delay = %.2d', n, d, f_answer / 10(9), delay))
legend('Q1', 'Q2')

% Uncomment for the plot of Q1 and Q2 of the first molecule of the chain
only

```

```

figure;
plot(t, Q1(1,:), t, Q2(1,:), 'linewidth', 1.2);
xlabel('Time [s]')
ylabel('Q1, Q2 [%]')
title('Evolution in time of Q1 and Q2 of the first molecule of Butane',...
      ..'FontWeight', 'bold')
subtitle(sprintf('Simulation with n = %d, d = %.2d,..
      ..switch f = %d GHz', n, d, f_answer / 10^(9)))
legend('Q1', 'Q2')

%% Uncomment for the plot with the difference between the exact
%%exponential behaviour and the FEM approximation

% u = ones(size(t));
% t_step = 0.5 * 10^(-12);
% u(t >= t_step) = 0;    % Step from 0 to 1 at t=2
% figure;
%
% % Subplot 1: Step input
% subplot(3,1,1);
% plot(t, u, 'LineWidth', 1.2);
% ylim([-0.1 1.1]);
% title('Step Input');
% ylabel('Vin [V]');
% xlabel('Time [s]');
%
% % Subplot 2: exact solution RC
% e = exp(-((t - t_step) .* (t >= t_step)) / tau);
% %y_exact = (1 - exp(-(t - t_step).*(t >= t_step) / tau));
% subplot(3,1,2);
% plot(t, e, 'r', 'LineWidth', 1.2);
% title('Exact solution RC');
% ylabel('Vout [V]');
% xlabel('Time [s]');
%
% % Subplot 3: FEM solution
% vFEM = zeros(size(t)); % approximation
% vFEM(1) = 1;
% dt = t(2) - t(1);
% for k = 2:length(t)
%     vFEM(k) = vFEM(k-1) + dt / tau * (u(k-1) - vFEM(k-1));
% end
%
% subplot(3,1,3);

```

---

```

% plot(t, e, 'r', 'LineWidth', 1.2); hold on; % exact
% plot(t, vFEM, 'b--', 'LineWidth', 1.2); % approximation
% ylabel('Vout [V]');
% xlabel('Time [s]');
% legend('Exact solution', 'Approximation');
%
% % General title
% sgtitle('RC response to a step input: exact solution and FEM ..
    ..approximation', 'FontWeight', 'bold');

% %% Uncomment for the plot of the vmol and the charge distribution
%for every molecule figure of INPUT of first molecule and OUTPUT over time of all the
%molecule in the chain
figure;
for L = 1:2 : 2 * n
    subplot(n, 2, L)
    l = ceil(L / 2);
    plot(t, vMol_in_time(l,:), 'b', 'linewidth', 1.2);
    xlim([0 1.5 * delay])
    ylim([-1.5 1.5])
    xlabel('Time [s]')
    ylabel(sprintf('V mut %d [V]', l))
    title(sprintf('V %d MUT', l))

    subplot(n, 2, L + 1)
    plot(t, Q1(l,:), '--b', t, Q2(l,:), 'linewidth', 1.2)
    xlim([0 1.5 * delay]);
    ylim([0 1]);
    title(sprintf('Q1, Q2 %d MUT', l))
    xlabel('Time [s]');
    ylabel('Q1, Q2 [%]')
end

legend1 = legend('Q1', 'Q2');
set(legend1, 'Position', [0.85 0.85 0.05 0.05]);

sgtitle(sprintf('Output voltage and distribution of charge for BUTANE molecule \n..
    ..n = %d, f = %.2d GHz, d = %.2d nm \n Delay = %.2d ps ', n, f_answer / 10^(9),..
    ..d * 10^(9), delay * 10^(12)), 'fontweight', 'bold');

```

## A.2 bis\_ferrocene.m

Below the script for create a simulation of the molecular wire made of bisferrocene molecule.

```

%% BISFERROCENE MOLECULE

% script that calculate the transient response of the BISFERROCENE MOLECULE
% The BISFERROCENE MOLECULE are formed by 4 dots. Q1 and Q2 are used to
% transmit the information, so they are the polarizing position of the
% molecule, Q3 and Q4 are the position where the charges are stored when the
% molecule is in an unstable equilibrium point.
%Also the clock is considered in the simulation, so we can have different
% dynamic behaviour

clear all;
clc
close all;

molecule = 'bis-ferrocene';

%n = number of molecule in the chain
%t = time vector
n = 6;
t = 0 : 10−15 : 10*10−12;
timeF = length(t); %Final time instant

%f_answer = experimental value of the switch frequency of the molecule
%tau = time constant of the molecule (equivalent RC constant)
f_answer = 250 * 109; %Hz
tau = 1/(2*pi*f_answer); %s

%d = distance between the center point of the driver molecule and the MUT
%w = length of the molecule
d = 0.5 *(10−9); % meter
w = 0.5*(10−9); %meter

%UNCOMMENT if you want a visual representation of the molecule position
%in space (only Q1 and Q2 are shown, since Q3 and Q4 are in the y-axis)
%DrawChain (w,d,n)

%electric constants
e = 1.6 *10−19; %Coulomb -> charge of the electron
eps0 = 8.854187817* 10−12; %Faraday/m = electric permittivity in air
k = 1/(4*pi*eps0); %electrostatic constant

%position of the charge of the external driver molecule
r1_D = [0;w/2];
r2_D = [0;-w/2];

```

---

```

%distribution of charge in the driver = polarized molecule
Q1_D = 0.9; %C
Q2_D = 0.1; %

% vin_clock = input voltage of the clock, the orthogonal contribution
% It can be always the same or it can varies in time
vin_clock = -2 * ones (length(t),1);

%Q1/2_0 = initial value of the charge of the molecule in the chain
%Q3 and Q4 are considered NULL in polarized state
Q1_0 = 0.5*ones(n,1);
Q2_0 =0.5*ones(n,1);

%initial voltage of the molecole
vmol_0 =zeros(n,1); %since Q1_0 = Q2_0 = 0.5*e

%Q1/2 = matrix n X timeF -> n= n° of molecule in the chain; time instant
%so in i-row there is the evolution in time of Q1/2 of the i-molecule
%in i-column we have the photo of how the Q1/2 are at time i
%Q3/4 = at initial state are considered null
Q1 = zeros(n,timeF);
Q2 = zeros(n,timeF);
Q3 = zeros(n,timeF);
Q4 = zeros(n,timeF);
Q1(:,1) = Q1_0;
Q2(:,1) = Q2_0;

%vMol_in_time= vector with vmol at every time instant
vMol_in_time= zeros(n,length(t));
vMol_in_time(:,1) = vmol_0;

%x = position of every molecule(same for Q1 and Q2)
%y = position of Q1 -> always equal to w/2
%y = position of Q2 -> always equal to -w/2
x = d : d : n*d;

%calculation of vin (constant for every molecule since we impose that
%Q1_D and Q2_d are constant
%vin = vector [n] where vin(i) = vin_D for molecule i
vin_D = zeros(n,1);
vin_D = k*(Q1_D./x + Q2_D./sqrt(x.^2+w^2) - ..
          ..Q1_D./sqrt(x.^2+w^2) -Q2_D./x);

```

```

%function that will compute the evolution in time of the molecule
[Q1, Q2, Q3,Q4,vMol_in_time]= EvolutionInTime (t, vin_D, Q1, Q2,..
    ..vMol_in_time,molecule,n,d,w,tau, vin_clock, Q3, Q4);

% COMMENT to skip the plot of the distribution of charge in the last molecule
t_T = 0: 0.6 * 10(-16): t(end);
vmol_T = interp1(t,vMol_in_time(n,:),t_T);
Q1_T = interp1(t,Q1(n,:),t_T);
Q2_T = interp1(t,Q2(n,:),t_T);
%Find delay of the convergency in the last molecule
delay = FindDelay(t_T,Q1_T(:),Q2_T(:));

%%UNCOMMENT for the plot of Q1 and Q2 of the last molecule of the chain
figure;
plot(t_T,Q1_T,t_T,Q2_T,'linewidth',1.2);
legend('Q1','Q2')
xlabel('Time [s]')
ylabel('Q1, Q2 [%]')
title('Evolution in time of Q1 and Q2 of the last bis-ferrocene',...
    ..'FontWeight', 'bold')
subtitle (sprintf('Simulation with n = %d, d = %.1d nm,..
    ..switch f= %d GHz \n Delay = %.2d', n, d* (10)9,f_answer/10(9),delay))

figure;
plot(t,Q1(1,:),t,Q2(1:,:),'linewidth',1.2);
legend('Q1','Q2')
xlabel('Time [s]')
ylabel('Q1, Q2 [%]')
title('Evolution in time of Q1 and Q2 of the first bis-ferrocene',...
    ..'FontWeight', 'bold')
subtitle (sprintf('Simulation with n = %d, d = %.1d nm,..
    ..switch f= %d GHz', n, d * (10)9,f_answer/10(9)))

%% UNCOMMENT for the plot in time of the output voltage and the distribution
%of charge of all the molecule in the bis-ferrocene molecule

figure;
for L=1:2 : 2*n

    subplot(n,2,L)
    l = ceil(L/2);
    plot(t,vMol_in_time(l,:), 'b', 'linewidth',1.2);
    xlim([0 1.5*delay]);
    xlabel('Time [s]')

```

```

ylabel(sprintf('V mut%d [V]',1))
title (sprintf('V %d MUT',1))

subplot(n,2,L+1)
%uncomment to see also Q3 and Q4
%plot(t,Q1(1,:),t,Q2(1,:),t,Q3(1,:),t,Q4(1:,:),'linewidth',1.2)
plot(t,Q1(1,:),t,Q2(1:,:),'linewidth',1.2)
xlim([0 1.5*delay]);
ylim([0 1]);
title(sprintf('Q1,Q2 %d MUT',1))
xlabel('Time [s]');
ylabel('Q1,Q2 [%]')
end
legend1 = legend ('Q1','Q2');
set(legend1,...
    'Position',[0.85 0.85 0.05 0.05]);

sgtitle(sprintf('Output voltage and distribution of charge for
BIS-FERROCENE molecule \n n = %d, f = %.2d GHz, d = %.2d nm \n ..
..Delay = %.2d ps ',n,f_answer/10^(9),d*10^(9),delay*10^(12)),...
    ..'fontweight', 'bold');

```

### A.3 Majority\_voter.m

Script that allow for the simulation of the majority voter made of bisferrocene or butane.

```

%MAJORITY VOTER BUTANE/BISFERROCENE MOLECULE

% script that calculate the transient response of a majority voters made of
% MOLECULE that can be chosen, so with the contributions of the clock
% Input are 3 different molecule which are followed by a bisferrocene
% molecule that acts as a buffer between the input and the molecule that
% will "perform" the calculation
clear all;
clc
close all;

molecule = 'butane';
%molecule = 'bis-ferrocene';

```

```

%n = number of molecule in the chain
% t = time vector
%d = intermolecular distance
%w = length of the molecule = distance between dot1 and dot2
n = 4;
t = 0 : 10^(-15) : 5* 10^(-12); %every time step is 1fs -> t(i) = t_i fs
%f_answer = experimental value of the switch frequency of the molecule
%tau = time constant of the molecule (equivalent RC constant)
f_answer = 650 * 10^9;%Hz
tau = 1/(2*pi*f_answer); %s

timeF = length(t); %Final time instant
w = 0.5*(10^-9); %meter -> bis-ferrocene
%w = 0.45*(10^-9); %meter -> butane
%for majority voters -> For symmetric geometry we will consider w = d
d = w;

%UNCOMMENT for the visual representation of the molecule position in
%space
DrawMaj(w,d,n)

%electric constants
e = 1.6 *10^(-19); %Coulomb ->charge of one electron
eps0 = 8.854187817* 10^(-12); %Fardas/m = electrici permittivity in air
k = 1/(4*pi*eps0); %electrostatic constant

% a bis-ferrocene molecule is composed by 2 different molecule, each one
%is characterized by 2 charges
%The input for the majority voter need to be made of a bis-ferrocene
%or butane molecule to allow a correct computation of the majority voter,
%so we will use a matrix 2x2 where every element of the matrix represent
%one of the dot in the bis-ferrocene molecule
%a function is used to assign for every input the relative Q
%in = the 3 different input for the MV
in = [1 0 1];

%Q_in{i} = matrix 2x2 with the distribution of charge of input i
Q_in = [0 0 0 0; 0 0 0 0; 0 0 0 0];
for i = 1 : 3
    qin = DriverInput(in(i));
    Q_in(i,:) = qin;
end

```

---

```

%R_in{i} = position of the charges for the input i

R_in1 = [0 d/2; d d/2; 0 -d/2; d -d/2];
R_in2 = [2*d 5*d/2; 3*d 5*d/2; 2*d 3*d/2; 3*d 3*d/2];
R_in3 = [2*d -3*d/2; 3*d -3*d/2; 2*d -5*d/2; 3*d -5*d/2];
R_IN = {R_in1 R_in2 R_in3};

%Q1/2_0 = initial value of the charge of the molecule in the chain
%Q1_0 = zeros(n,1);
Q1_0 = 0.5*ones(n,1);%.*e;
Q2_0 = 0.5*ones(n,1);%.*e;

%initial voltage of the molecule
vmol_0 = zeros(n,1); %since Q1_0 = Q2_0 = 0.5*e

%Q1/2 = matrix n X timeF -> n= n° of molecule of ferrocene (it takes 2 to
%become a bisferrocene molecule in the chain; time instant so in
%i-row there is the evolution in time of Q1/2 of the i-molecule in
%i-column we have the photo of how the Q1/2 are at time i
Q1 = zeros(n,timeF);
Q2 = zeros(n,timeF);
Q3 = zeros(n,timeF);
Q4 = zeros(n,timeF);
Q1(:,1) = Q1_0;
Q2(:,1) = Q2_0;

%vMol_in_time= vector with vmol at every time instant
vMol_in_time= zeros(n,length(t));
vMol_in_time(:,1) = vmol_0;

%Positions of the molecule in the main chain
%x = position of every molecule(same for Q1 and Q2)
%y = position of Q1 -> always equal to w/2
%y = position of Q2 -> always equal to -w/2
x = 2*d : d : (n + 1)*d;

%matrix of the position of the charge Q1 of the molecule in the chain
% first column = x
% second column = z
R1= [x' d/2*ones(length(x),1)];
R2= [x' -d/2*ones(length(x),1)];

%Calculation of all 3 Vin (considered constant at every time instant) for
%every molecule in the chain

```

```
% vin_D = matrix where vin_D(i,j) = vin of input i for molecule j
vin_D = zeros(3,n);
for i = 1 : 3
    R_in = R_IN{i};
    for p = 1 : n
        vtot1 = 0;
        vtot2 = 0;
        for j = 1:4
            vtot1 = vtot1 + Q_in(i,j)./norm(R_in1(j,:) - R1(p,:));
            vtot2 = vtot2 + Q_in(i,j)./norm(R_in1(j,:) - R2(p,:));
        end
        vin_D(i,p) = k * (vtot1 - vtot2);
    end
end

end
Vtot_D(:) = vin_D(1,:) + vin_D(2,:) + vin_D(3,:);

% vin_clock = input voltage of the clock, the orthogonal contribution
% It can be always the same or it can varies in time
vin_clock = 2 * ones (length(t),1);

%function that will compute the evolution in time of the molecule
if strcmp(molecule, 'bis-ferrocene')
    [Q1, Q2, Q3,Q4,vMol_in_time]= EvolutionInTime (t, vin_D, Q1, Q2,..
        ..vMol_in_time, molecule,n, d,w,tau, vin_clock, Q3, Q4);
end
if strcmp(molecule, 'butane')
    [Q1,Q2,vMol_in_time] = EvolutionInTime (t,vin_D, Q1, Q2,..
        ..vMol_in_time, molecule,n, d, w ,tau);
end

end

%% PLOTTING

%xt = new point for time (should be different and less than before
t_T = 0: 6*10^(-16) : t(end);
%extracting the linear polinomial for the previous point, then it will
%use the linear polinomial to find the relative vmol for the new xt
vmol_T = interp1(t,vMol_in_time(n,:),t_T);
QA_T = interp1(t,Q1(n-1,:),t_T);
QB_T = interp1(t,Q2(n-1,:),t_T);
QC_T = interp1(t,Q1(n,:),t_T);
QD_T = interp1(t,Q2(n,:),t_T);
%Find delay of the convergency in the last molecule
delay = FindDelay(t_T,QC_T,QD_T)
```

```

figure;
plot(t_T,QA_T,t_T,QB_T,t_T,QC_T,t_T,QD_T,'linewidth',1.2);
legend('Q^3_1','Q^3_2','Q^4_1','Q^4_2')
xlabel('Time [s]')
ylabel('Q^3_1, Q^3_2, Q^4_1, Q^4_2 [%]')
if strcmp(molecule, 'butane')
    title('Majority voter BUTANE: evolution in time of the last cell',...
        ..'FontWeight', 'bold')
else
    title('Majority voter BISFERROCENE: evolution in time of the last cell',...
        ..'FontWeight', 'bold')
end
subtitle (sprintf('Switch frequency = %d GHz \n..
    ..Delay = %.2d', f_answer/(10^9),delay))
in

%Output of the majority voter in the command window, if the output is
%different than the expected one, last = 9
last = Output (Q1(n-1,end),Q1(n,end),Q2(n-1,end),Q2(n,end))

```

## A.4 EvolutionInTime.m

Function used in simulationButane.m, bis\_ferrocene.m and Majority\_voter.m for the time dependent part of the simulation.

```

function varargout = EvolutionInTime (t, Vtot_D, Q1,Q2,vMol_in_time, ..
    ..molecule,n,d,w,tau, varargin)
%function that allows to calculate the evolution in time of the
%molecule, once the voltage generated by the driver on every
%molecule is already calculate
% OUT = Q1, Q2 if molecule = 'butane'
% OUT = Q1, Q2 Q3,Q4 if molecule = 'bis-ferrocene'
%t = time vector
%Vtot_D = vector [n] with the voltage generated by the 2 charges of the
%driver(considered STATIONARY)
%Q1, Q2 = matrix [ n x length(t)] for the evolution in time of Q1 and Q2
%for the molecule in the chain
%vMol_in_time = matrix [n x length(t)] for evolution in time of the voltage
%in a molecule
%molecule = 'butane' or 'bis-ferrocene'
%n = number of molecule in the chain
%d = intramolecular distance (equally spaced)
#w = intermolecular distance

```

```

%tau = time constant related to the switch frequency of a molecule
% if molecule = 'bis-ferrocene' -
%varargin{1} = clock voltage
%varargin{2} = Q3 matrix [n x timeF] evolution in time of Q3
%varargin{3} = Q4 matrix [n x timeF] evolution in time of Q4

%electric constants
e = 1.6 *10(-19); %Coulomb -> charge of the electron
eps0 = 8.854187817* 10(-12); %Faraday/m = electric permittivity in air
k = 1/(4*pi*eps0); %electrostatic constant
timeF = length(t); %number of time samples

% Initialization of optional variable
vin_clock = []; % Defined only for bis-ferrocene

if strcmp(molecule, 'bis-ferrocene')
    %define vin_clock
    if length(varargin) >= 3
        vin_clock = varargin{1};
        Q3 = varargin {2};
        Q4 = varargin {3};

    end

    % vin_clock = input voltage of the clock, the orthogonal contribution
    % It can be always the same or it can varies in time

    filename1 = 'ck1.txt'; % clock = -2 V/m
    filename2 = 'ck2.txt'; % clock = 0 V/m
    filename3 = 'ck3.txt'; % clock = +2 V/m

    % Import file from databse to derive tha transcharacteristic of
    % bis-ferrocene
    [Vin_q1,Q1_q1,Q2_q1,Q3_q1,Q4_q1]= VACT_bisf(filename1);
    [Vin_q2,Q1_q2,Q2_q2,Q3_q2,Q4_q2]= VACT_bisf(filename2);
    [Vin_q3,Q1_q3,Q2_q3,Q3_q3,Q4_q3]= VACT_bisf(filename3);
    %Vin_1x = 'x' of the transcharacteristic (see function VACT) for
    %relativeclock
    %Q1_qx, Q2_qx, Q3_x, Q4_x = 'y' of the transcharacteristic
end

if strcmp(molecule, 'butane')
    % Import file from databse to derive tha transcharacteristic of butane
    filename = 'only.txt';

```

---

```

[V_vact,Q1_vact,Q2_vact]= VACT(filename);
%V_vact = 'x' of the transcharacteristic (see function VACT)
%Q1_vact, Q2_vact = 'y' of the transcharacteristic (see functiojn VACT)
end

%x = position of every molecule(same for Q1 and Q2)
%y = position of Q1 -> always equal to w/2
%y = position of Q2 -> always equal to -w/2
x = d : d : n*d;

%calculation of distances
%r11 = matrix nxn where r(i,j) = distance from Q1/2_i from all Q1/2_j
%r12 = matrix nxn where r(i,j) = distance from Q1/2_i from all Q2/1_j
r11 = zeros(n,n);
r12 = zeros(n,n);
for j=1 : n
    rmut = [x(j),w/2];
    for i = 1 : n
        r11(j,i) = norm(rmut-[x(i),w/2]);
        r12(j,i) = norm(rmut-[x(i),-w/2]);
    end
end

%evaluation for every time instant of the charges calculating the FEM of
%the molecule

%find vin = divided in 4 contributions + 2 from the driver
%q1r1 = effect of Q1(ti) on dot1 of mut
%q1r2 = effect of Q1(ti) on dot 2 of mut
%q2r1 = effect of Q2(ti) on dot1 of mut
%q2r2 = effect of Q2(ti) on dot2 of mut
q1r1 = zeros(timeF,1);
q1r2 = zeros(timeF,1);
q2r1 = zeros(timeF,1);
q2r2 = zeros(timeF,1);
vin = zeros (timeF,1);
for i=1 : length(t)-1

    for j=1 : n
        %Mask = logical vector that will determine which molecule are
        %considered when calculating the effect on another molecule
        Mask = ones(n,1,"logical");

        %index of the mask of the MUT = 0 to exclude the molecule from the

```

```
%calculation
Mask(j) = 0;
q1r1(j,i) = sum((Q1(Mask,i) ./ r11(Mask,j)), 'omitnan');
q1r2(j,i) = sum((Q1(Mask,i) ./ r12(Mask,j)), 'omitnan');
q2r1(j,i) = sum((Q2(Mask,i) ./ r12(Mask,j)), 'omitnan');
q2r2(j,i) = sum((Q2(Mask,i) ./ r11(Mask,j)), 'omitnan');

vin(j,i) = Vtot_D(j) + k*(q1r1(j,i)+q2r1(j,i) -..
    ..(q1r2(j,i) + q2r2(j,i)));

%final calculation of the output of molecule j
vMol_in_time(j,i+1) = vMol_in_time(j,i) + ((t(i+1)-t(i))/tau)*..
    ..(vin(j,i)-vMol_in_time(j,i));

if strcmp(molecule, 'bis-ferrocene')
    %finding Q1 and Q2 of t_i+1 with VACT bis-ferrocene
    for p=1 : length(Vin_q1)
        if vin_clock(i) == -2
            if abs(vMol_in_time(j,i+1)*e - Vin_q1(p)) < ..
                ..(Vin_q1(2)-Vin_q1(1))
                Q1(j,i+1) = Q1_q1(p);%*e;
                Q2(j,i+1) = Q2_q1(p);%*e;
                Q3(j,i+1) = Q3_q1(p);%*e;
                Q4(j,i+1) = Q4_q1(p);%*e;
            end
        end
        if vin_clock(i) == 0
            if abs(vMol_in_time(j,i+1)*e-Vin_q2(p)) < ..
                ..(Vin_q2(2)-Vin_q2(1))
                Q1(j,i+1) = Q1_q2(p);%*e;
                Q2(j,i+1) = Q2_q2(p);%*e;
                Q3(j,i+1) = Q3_q2(p);%*e;
                Q4(j,i+1) = Q4_q2(p);%*e;
            end
        end
        if vin_clock(i) == 2
            if abs(vMol_in_time(j,i+1)*e-Vin_q3(p)) < ..
                ..(Vin_q3(2)-Vin_q3(1))
                Q1(j,i+1) = Q1_q3(p);%*e;
                Q2(j,i+1) = Q2_q3(p);%*e;
                Q3(j,i+1) = Q3_q3(p);%*e;
                Q4(j,i+1) = Q4_q3(p);%*e;
            end
        end
    end
end
```

```

        end
    end
    if strcmp(molecule, 'butane')
        %finding Q1 and Q2 of t_i+1 with VACT butane
        for p=1 : length(V_vact)
            if abs(vMol_in_time(j,i+1)*e-V_vact(p)) < ..
                ..(V_vact(2)-V_vact(1))
                Q1(j,i+1) = Q1_vact(p);%*e;
                Q2(j,1+i) = Q2_vact(p);%*e;
            end
        end
    end
end
end
end
if strcmp(molecule, 'bis-ferrocene')
    varargout = {Q1, Q2, Q3, Q4,vMol_in_time*e}; % 5 Output
elseif strcmp(molecule, 'butane')
    varargout = {Q1, Q2,vMol_in_time*e}; % 3 Output
end
end
end

```

## A.5 FindDelay.m.m

Function used for the calculation of the delay, called in every simulation.

```

function delay=FindDelay(t,Q1,Q2)
%Function that given Q1 in time and Q2 in time and the time vector, can
%calculate the time needed fo reaching convergency

%t = vector with time instant where Q1 and Q2 are given
%Q1 = vector with time evolution of Q1
%Q2 = vector with time evolution of Q2

%check if the convergency of the molecule is reached
flag_conv=0;
conv_reached = 0; %flag to stop the searching for convergency
conv = 0.005;%* e; %convergency limit

c= 1;
delay = 99;
% number of following sample less than convergency to ensure convergency is
% actually reached
n_stable = 10;
%summing the input, if sum bigger or equal to 2 -> output shall be 1
while (conv_reached < n_stable) && (t(c)< t(end))

```

```

c = c+1;
%if to understand if i have the same value of charges as in the
%previous cycle
%check between 2 following time instant + check between this time
%instant and the last one, to assure that it is not just a plateau
if (abs(Q1(c) - Q1(c-1)) < conv) && (abs(Q2(c) - Q2(c-1)) < conv) ..
    ..&& (abs(Q1(c) - Q1(end)) < conv) && (abs(Q2(c) - Q2(end)) < conv)
    if (flag_conv == 1)
        % if also at previous time the convergency was reached
        conv_reached = conv_reached + 1;
        %then number of stable sample increase
    end
    flag_conv = 1;
    % flag that allow to know if we enter in the convergency if previous time

else
    %if i did not enter in the convergency if, i erase the flag and
    %start to convergency counter to 0
    flag_conv = 0;
    conv_reached = 0;
end
%if i enter in the convergency if 3 times -> found index of the delay,
%exit the while
if (conv_reached == n_stable)
    delay = t(c - n_stable);
end
end
end
end

```

## A.6 VACT.m and VACT\_bisf.m

Function that allows the construction of the VACT, VACT is for the butane molecule, while VACT\_bisf is for the bisferrocene.

```

function [Vin_q,Q1_q,Q2_q]= VACT(filename)
%prende i dati dalla caratterizzazione della molecola e trova la
%transcaratteristica della molecola che lega la tensione della molecola
%alla quantità di carica Q1 e Q2 presente nei due dot

%import database with dedicated function

[Vin_f,Q1_f,Q2_f] = import_file(filename);

%linear interpolation of the data

```

---

```

% Q1_q = Q1 of the new query points
%xq = query points

Vin_q = linspace(-2,2,5000);
Q1_q = (interp1(Vin_f,Q1_f,Vin_q,"spline"))';
Q2_q = (interp1(Vin_f,Q2_f,Vin_q,"spline"))';

%UNCOMMENT to see plot of the transcharacteristics of the molecule

figure;
plot(Vin_f,Q1_f,'+',Vin_f,Q2_f,'*');
xlim([-2;2]);
hold on

plot(Vin_q,Q1_q,'m',Vin_q,Q2_q)
legend('Q1 exact point from database ', 'Q2 exact point from database',...
       'Q1 interpolated point','Q2 interpolated point');
title ('VACT')
xlabel('V_{MUT}')
ylabel('Q1_{MUT}, Q2_{MUT}')
```

```

function [Vin_q,Q1_q,Q2_q,Q3_q,Q4_q]= VACT_bisf(filename)
%prende i dati dalla caratterizzazione della molecola e trova la
%transcaratteristica della molecola che lega la tensione della molecola
%alla quantità di carica Q1 e Q2 presente nei due dot

%import database with dedicated function

% filename = 'ck1.txt';
[Vin_f,Q1_f,Q2_f,Q3_f,Q4_f] = importfile(filename);

%linear interpolation of the data
% Q1_q = Q1 of the new query points
%xq = query points

Vin_q = linspace(-10,10,5000);
Q1_q = (interp1(Vin_f,Q1_f,Vin_q,"spline"))';
Q2_q = (interp1(Vin_f,Q2_f,Vin_q,"spline"))';
Q3_q = (interp1(Vin_f,Q3_f,Vin_q,"spline"))';
Q4_q = (interp1(Vin_f,Q4_f,Vin_q,"spline"))';

%UNCOMMENT to see the plot of the transcharacteristics of the molecule

% figure;
% plot(Vin_f,Q1_f,'+',Vin_f,Q2_f,'*',Vin_f,Q3_f,'o',Vin_f,Q4_f,'x');
```

```

% xlim([-2;2]);
ylim([0; 1]);
hold on

plot(Vin_q,Q1_q,'m',Vin_q,Q2_q,Vin_q,Q3_q,Vin_q,Q4_q)

legend('Q1 exact point from database ', 'Q2 exact point from database',
    ..., 'Q3 exact point from database','Q4 exact point from database',
    ...'Q1 interpolated point','Q2 interpolated point',...,
    'Q3 interpolated point','Q4 interpolated point');
title ('VACT')
xlabel('V_{MUT}')
ylabel('Q1_{MUT}, Q2_{MUT}')

```

## A.7 AnalysisButane\_different\_d\_n\_f\_switch.m

This is the script for the study of the delay as a function of the number of molecule, on the intermolecular distance and of the frequency switch.

```

%%
clc;
clear;
close all;

% Definition of analysis parameters
% Variation of the number of molecules
n_values = 3:1:30;
% Variation of the distance
d_values = linspace(0.3e-9, 0.63e-9, 30); % Variation of the distance
% Switching frequency expressed in GHz
f_switch_values = 450e9:25e9:900e9; % Switching frequency

F_switch_ref = 640e9;
N_ref = 8; %o anche 10
d_ref = 0.5e-9;

%Suggestions: for quick analysis, use only the section that you need
%otherwise there will be a long computation time
%% Analysis for delay as a function of n
delay_n = (-1)*ones(1,length(n_values));

for i=1 : length(n_values)
    % Call with fixed values of d and f_switch
    delay_n(i) = simulationButane(n_values(i), d_ref, F_switch_ref);

```

```
end

% Plot results for delay vs n
figure;
plot(n_values,delay_n)
xlabel('Number of molecules (n)');
ylabel('Delay (fs)');
title('Delay as a function of the number of molecules');
subtitle(sprintf('f_switch = %d GHz; d = %.2d',F_switch_ref/10^(9),d_ref));
grid on;

%% Analysis for delay as a function of d
delay_d = (-1)*ones(1,length(d_values));
for i=1:length(d_values)
    % Call with fixed values of n and f_switch
    delay_d(i) = simulationButane(N_ref, d_values(i), F_switch_ref);
end

% Plot results for delay vs d
figure;
plot(d_values,delay_d, '-o');
xlabel('Distance (nm)');
ylabel('Delay (ps)');
title('Delay as a function of distance');
subtitle(sprintf('f_swith = %d; n = %d',F_switch_ref/10^(9),N_ref));
grid on;

%% Analysis for delay as a function of f_switch
delay_f = (-1)*ones(1,length(f_switch_values));
for i=1:length(f_switch_values)
    delay_f(i) = simulationButane(N_ref, d_ref, f_switch_values(i));
end

% Plot results for delay vs f_switch
figure;
plot(f_switch_values/(10^(9)), delay_f);
xlabel('Switching frequency (GHz)');
ylabel('Delay (s)');
title('Delay as a function of switching frequency');
subtitle(sprintf('n = %d; d = %.2d', N_ref, d_ref));
grid on;
```



# Bibliography

- [1] ITRS experts, "International Technology Roadmap for Semiconductors", 2015 report, 2015.
- [2] Y. Ardesi, G. Turvani, and G. Piccinini, *SCERPA Simulation of Clocked Molecular Field-Coupling Nanocomputing*, IEEE Transactions on Nanotechnology, 2020.
- [3] Y. Ardesi, A. Gaeta, G. Beretta, G. Piccinini, and M. Graziano, "Ab initio Molecular Dynamics Simulations of Field-Coupled Nanocomputing Molecules," *JICS. Journal of Integrated Circuits and Systems*, vol. 16, no. 1, pp. 1–8, 2021, doi: [10.29292/jics.v16i1.474](https://doi.org/10.29292/jics.v16i1.474).
- [4] Y. Ardesi, R. Wang, G. Turvani, G. Piccinini, and M. Graziano, "SCERPA: A Self-Consistent Algorithm for the Evaluation of the Information Propagation in Molecular Field-Coupled Nanocomputing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2749–2760, 2020. doi:10.1109/TCAD.2019.2960360.
- [5] R. Listo, F. Ravera, G. Beretta, Y. Ardesi, G. Piccinini, and M. Graziano, "Unveiling Charge Dynamics in Molecular Field-Coupled Nanocomputing," *2024 IEEE 24th International Conference on Nanotechnology (NANO)*, Gijon, Spain, 2024, pp. 424–429, doi: 10.1109/NANO61778.2024.10628616.