

**E**VOLVING  
**N**ETWORKS  
**T**O  
**H**AVE  
**I**NTELLIGENCE  
**R**EALIZED  
**A**T  
**N**ANOSCALE

## Graduation committee

Prof.dr. J.N. Kok	Chairperson and secretary, University of Twente
Prof.dr.ir. W.G. van der Wiel	Supervisor, University of Twente
Prof.dr.ir. H.J. Broersma	Supervisor, University of Twente
Prof.dr.ir. B.J. Geurts	University of Twente
Prof.dr.ing. A.J.H.M. Rijnders	University of Twente
Prof.dr. P. Hadley	Graz University of Technology, Austria
Prof.dr. P.H.E. Tiesinga	Radboud University, Nijmegen

## Acknowledgement

The research described in this thesis was performed at the faculty of Electrical Engineering, Mathematics and Computer Science, and the MESA+ Institute for Nanotechnology at the University of Twente. We acknowledge financial support from the MESA+ Institute for Nanotechnology and the CTIT Institute for ICT research, as well as the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement No. 317662.

*Evolving networks to have intelligence realized at nanoscale (ENTHIRAN)*

Copyright © Celestine Preetham Lawrence, 2018.

Printed by Ipskamp Drukkers, Enschede.

ISSN: 2589-4730 (IDS PhD thesis Series No. 18-464)

ISBN: 978-90-365-4547-1

DOI: 10.3990/1.9789036545471

# **EVOLVING NETWORKS TO HAVE INTELLIGENCE REALIZED AT NANOSCALE**

DISSERTATION

to obtain  
the degree of doctor at the University of Twente,  
on the authority of the rector magnificus,  
prof.dr. T.T.M. Palstra,  
on account of the decision of the graduation committee,  
to be publicly defended  
on Wednesday the 16<sup>th</sup> of May 2018 at 14.45 hours

by

**Celestine Preetham Lawrence**

born on the 6<sup>th</sup> of April 1992  
in Chennai, India

This dissertation has been approved by:

Supervisor: Prof.dr.ir. W.G. van der Wiel

Supervisor: Prof.dr.ir. H.J. Broersma

## ACRONYMS

SET	Single-electron tunnelling
NP	Nanoparticle (of gold)
GA	Genetic algorithm
NDR	Negative differential resistance
MC	Monte Carlo
MF	Mean-field
IQ	Intelligence quotient
EIM	Evolution in materio
NASCENCE	NANoScaLe Engineering for Novel Computation using Evolution

## RECTIFIER FUNCTION

$$[x] \equiv \max(0, x)$$

## THRESHOLDING FUNCTION

$$(x|y) \equiv [x - y] - [-x - y]$$

## NOTATION OF VARIABLES

Scalar  $x$  is italic, vector  $\boldsymbol{x}$  is bold italic and matrix  $\mathbf{x}$  is sans-serif bold. An exception is the SET network capacitance matrix  $\mathbf{C}$ , so that  $\mathbf{C}_{rr'}$  (element at row  $r$  and column  $r'$ ) is distinguishable from the mutual capacitance  $C_{rr'}$ .

# CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>1</b>
1.1	OUTLINE .....	3
<b>2</b>	<b>SINGLE-ELECTRON TUNNELLING IN NANOPARTICLE NETWORKS.....</b>	<b>7</b>
2.1	SET NETWORK MODEL OF A NP CLUSTER .....	7
2.1.1	<i>Free energy of a tunnel event .....</i>	<i>8</i>
2.1.2	<i>Tunnelling rates.....</i>	<i>9</i>
2.1.3	<i>Charge stability diagram.....</i>	<i>10</i>
2.1.4	<i>Charge dynamics at critical potential.....</i>	<i>12</i>
2.2	SIMULATION METHODS .....	14
2.2.1	<i>Monte Carlo method.....</i>	<i>15</i>
2.2.2	<i>Novel mean-field approximation.....</i>	<i>16</i>
2.2.3	<i>Comparison of Monte Carlo and mean-field simulations.....</i>	<i>18</i>
2.3	ILLUSTRATIVE SIMULATIONS .....	19
2.3.1	<i>Negative differential resistance (NDR).....</i>	<i>19</i>
2.3.2	<i>Bistability.....</i>	<i>20</i>
2.3.3	<i>Irregularity .....</i>	<i>20</i>
2.3.4	<i>Boolean logic in a regular SET network.....</i>	<i>21</i>
2.3.5	<i>Multiscale networks.....</i>	<i>22</i>
2.4	CONCLUSION .....	22
<b>3</b>	<b>FABRICATION OF NP CLUSTERS .....</b>	<b>25</b>
3.1	PATTERNING ELECTRODES ON A CHIP.....	25
3.2	NP TRAPPING BY SELF-ASSEMBLY .....	27
3.2.1	<i>Dielectrophoresis.....</i>	<i>27</i>
3.3	CONCLUSION .....	30
<b>4</b>	<b>EXPERIMENTS ON NP CLUSTERS.....</b>	<b>31</b>
4.1	COULOMB BLOCKADE FEATURES .....	32
4.2	CHANNEL LENGTH AND DIMENSIONALITY OF ELECTRON TUNNELLING .....	33
4.3	ABUNDANCE OF BOOLEAN FUNCTIONALITY (LOGIC AND MEMORY) .....	35
4.4	EVOLUTION OF BOOLEAN LOGIC GATES.....	40
4.4.1	<i>Fitness score .....</i>	<i>40</i>
4.4.2	<i>Genetic algorithm (GA).....</i>	<i>41</i>
4.4.3	<i>Universal evolvability in a compact device .....</i>	<i>42</i>
4.5	GA SEARCH CONVERGENCE SPEED AND OPTIMA .....	43
4.6	ROBUSTNESS OF THE DEVICE .....	45
4.6.1	<i>Combinatorial logic functionality.....</i>	<i>45</i>
4.6.2	<i>Input voltage error tolerance.....</i>	<i>46</i>

4.6.3	<i>Control voltage sensitivity</i> .....	46
4.6.4	<i>Stability over reconfigurations and thermal cycling</i> .....	48
4.6.5	<i>Stability over days</i> .....	49
4.7	MULTI-OUTPUT BINARY FUNCTIONALITY .....	49
4.8	CONCLUSION .....	49
<b>5</b>	<b>NP INTERNET</b> .....	<b>53</b>
5.1	LARGE-SCALE SMALL-WORLD ARCHITECTURE.....	53
5.2	SET NETWORK MODEL.....	55
5.3	SIMULATIONS OF NP INTERNETS AT INTERESTING WIRING CONFIGURATIONS .....	56
5.3.1	<i>Diversity in current-voltage relation</i> .....	57
5.3.2	<i>Inhibitory mechanism</i> .....	59
5.4	PATTERN RECOGNITION .....	62
5.4.1	<i>Abundance of functionality</i> .....	62
5.5	CONCLUSION .....	64
<b>6</b>	<b>IQ OF A PHYSICAL SYSTEM</b> .....	<b>65</b>
6.1	DEFINING INTELLIGENCE .....	65
6.1.1	<i>IQ factors of a table of relations</i> .....	67
6.2	IQ TEST FOR BOOLEAN LOGIC IN A NANOMATERIAL CLUSTER .....	68
6.3	CONCLUSION .....	70
<b>7</b>	<b>OUTLOOK</b> .....	<b>71</b>
7.1	EVOLUTION IN MATERIO .....	71
7.2	EVOLUTION AND THE BRAIN .....	72
7.3	NEUROEVOLUTION IN NANOMATERIO .....	73
7.4	OVERVIEW .....	73
<b>A</b>	<b>APPENDIX</b> .....	<b>76</b>
1)	<i>Mathematica code</i> .....	76
2)	<i>Numerical method to solve for equilibrium point of MF approximation</i> .....	76
3)	<i>Relative error</i> .....	77
4)	<i>Netlist of disordered network</i> .....	77
5)	<i>Netlist of 4×4 grid network</i> .....	78
6)	<i>Netlist of bistable circuit</i> .....	78
7)	<i>SIMON netlist of multiscale network</i> .....	78
8)	<i>Fabrication details</i> .....	79
9)	<i>Fitness quality</i> .....	79

## ACKNOWLEDGEMENTS

## SUMMARY IN DUTCH AND TAMIL

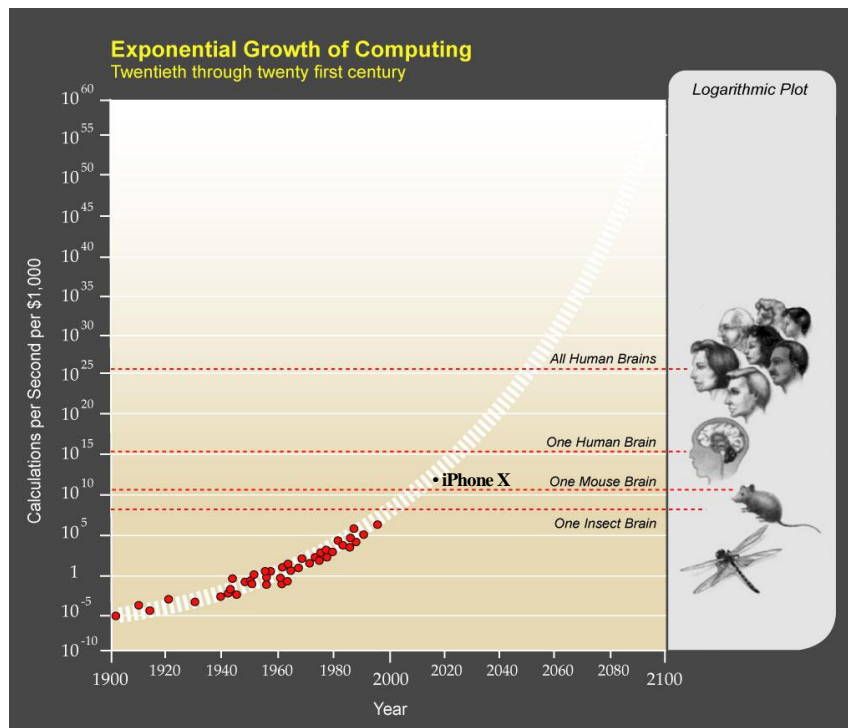




# 1 Introduction

“Hi, I’m Chitti the robot. Speed 1 terahertz, Memory 1 zettabyte.”

Those are the opening lines of a cheeky humanoid robot in the Indian Tamil movie ENTHIRAN [Shankar2010], a science fiction blockbuster of the year 2010. Moving forward, in the year 2100, such a fantastic scenario may actually be reality if we continue with the exponential growth of computing. See the infographic below to get an impression, adapted from [Kurzweil2010]. Its prediction is still valid in 2018, considering that Apple’s iPhone X is capable of up to  $6 \times 10^{11}$  operations per second.



Infographic sourced from Wikimedia Commons, by courtesy of Ray Kurzweil and Kurzweil Technologies, Inc.

Although what people consider artificial general intelligence is nowhere close to sight yet, machines with artificial neural networks capable of deep learning [LeCun2015,Schmidhuber2015] are getting closer to humankind. The latest smartphones have apps that can translate speech input and narrate the scenery captured on camera, in real-time and with reasonable accuracy. In other ‘playing’ fields, machines of Google’s DeepMind have achieved superhuman level in Atari 2600 video games [Mnih2015] and the classic board game Go [Silver2016]. DeepMind’s AlphaGo defeating the human world champion was breaking news because it is a game that requires a great deal of intuition [Nielsen2016] to choose winning moves out of possibilities that branch out to astronomically large numbers. However, to gain these victories, deep learning machines are relatively energy hungry. AlphaGo machine is made of 4 tensor processing units (TPUs, [Jouppi2017]) that consume  $\sim 40$  W each. In comparison, a human brain consuming  $\sim 20$  W on average is capable of way more multitasking!

In the coming few years, we shall see further developments in application-specific ICs [Han2016,Chen2017] to make more efficient deep learning machines. In fact, the Neural Engine in Apple's iPhone X implementing Face ID is a baby step in that direction. But to continue the exponential growth of computing, we may need a paradigm shift and come up with truly neuromorphic architectures [Gomes2017].

At the start of my PhD research, IBM released a ground-breaking chip called TrueNorth that could simulate artificial neural networks using a million spiking-neuron integrated circuit with a scalable communication network and interface [Merolla2014]. Their thumb-sized chip could perform multi-object detection and classification with  $400 \times 240$  pixel video input at 30 fps, while consuming a paltry 63 mW of energy. The reason for such great energy efficiency is because the system exchanges information by spikes of electronic conduction, similar in principle to spikes of ionic conduction in biological neural networks. However, training methods to fully exploit this kind of a system based on spiking neurons are still in development [Abbott2016,Eliasmith2012,Esser2015].

Okay, so far so good. But how can we push the envelope even further to reach the ultimate physical limits [Lloyd2000] of computing? Looking beyond neural networks, natural computers in general exploit the emergent properties and massive parallelism of interconnected networks of locally active components [Hopfield1982,Watts1998]. And it is the process of evolution that has resulted in intelligent systems that use energy efficiently, utilizing whatever physical processes are exploitable [Toffoli2004]. So instead of making circuits of functional units that have design rules to exclude physical processes such as capacitive crosstalk, it seems more natural to use matter in its designless form and exploit all possible physical processes for computation.

Miller and Downing [Miller2002] propose to use artificial evolution of complex physical systems for this purpose, and introduced the term 'evolution in materio'. They envision that a truly open-ended evolutionary process could exploit unknown physics and chemistry of the real world, leading to discoveries by serendipity. They suggested exploring materials like liquid crystals, conducting polymers and voltage controlled colloids due to their special properties. Initial experiments done on liquid crystals at the microscale to evolve a frequency discriminator proved successful [Harding2004], and this led to a growing interest in the field. The EU thus funded NASCENCE [Broersma2012], a project in which systems at the nanoscale could be engineered for novel computation using evolution. It was brandished as a 'High risk, High gain' project because of its extreme novelty, until then designless matter at the nanoscale had not been configured to perform computation.

Our initial experiment [Bose2015] with the evolution of a designless nanoparticle network into reconfigurable Boolean logic was successful and is considered to be a pioneering work in the field of 'evolution in nanomaterio' [Han2015]. It was also discovered that our system met the essential criteria for the physical realization of neural networks and thus began my journey into neuroevolution. The

proposition of ‘neuroevolution in nanomaterio’ is explained in Figure 1.1. It is suggested to revisit this figure after completing Chapter 5 for a deeper understanding.

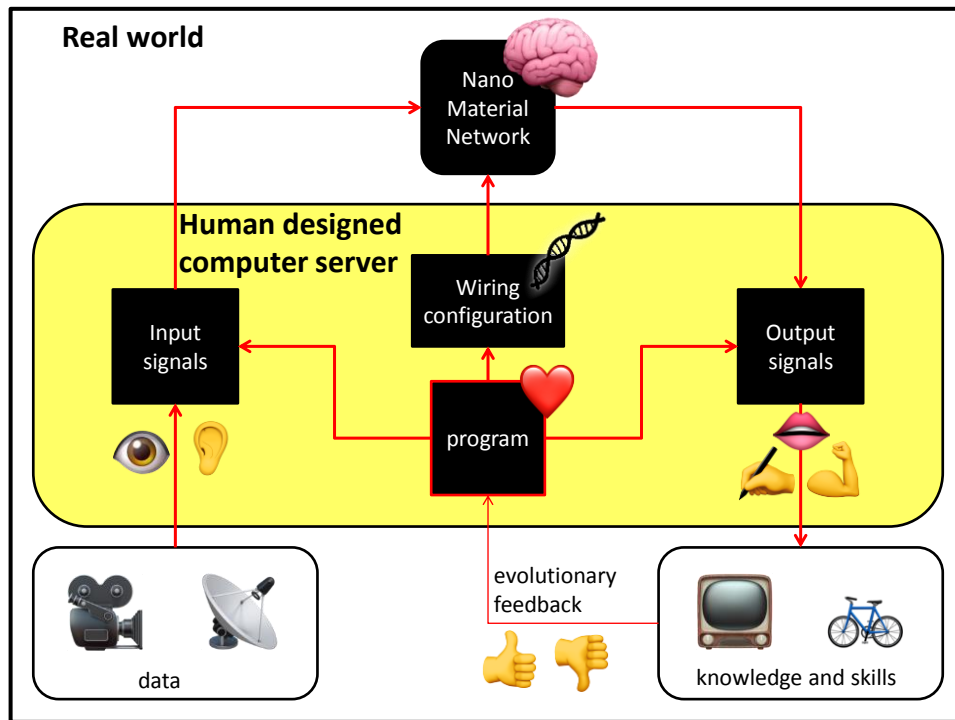


Figure 1.1. Neuroevolution in nanomaterio. An interconnectable network of nanomaterials is proposed to realize an intelligent mind by the evolution of its wiring configuration. A human designed computer serves as a body for the mind by: processing data (e.g. an image, audio time series) to input signals (e.g. input layer connectivity, time-varying voltages); providing knowledge and skills (e.g. object recognition, chat assistant, vehicle control) from output signals (e.g. charge states, time-varying currents); and running a genetic algorithm to optimize the wiring configuration.

## 1.1 Outline

A main objective of this thesis is to prove that an interconnectable network of nanoparticles can be tuned to perform pattern recognition by the evolution of its wiring configuration.

We start by describing the physics behind charge transport in a nanoparticle (NP) system by modelling it as a single-electron tunnelling (SET) network in Chapter 2. And, we propose a novel mean-field approximation to simulate the current-voltage relations of large-scale SET networks, much faster than a Monte Carlo method.

To build our NP system, in Chapter 3, we learn how to pattern electrodes on a silicon substrate and fill a cluster of NPs in between them. In our work, gold NPs coated with a molecular monolayer are trapped by dielectrophoresis (DEP). We apply DEP theory to derive parameters that result in effective trapping.

In Chapter 4, we look at experimental results on several independent NP clusters. From measurements across a large set of random voltage configurations, we study the abundance of

Boolean functions including logic gates and some kinds of memory. We see that using a genetic algorithm to optimize the voltage configuration produces even better results than a random search.

To scale up our system for greater functionality, in Chapter 5, we design a small-world architecture that interconnects several NP clusters by electronic switches whose wiring is configured by a computer server. We call this a ‘NP internet’ and show by means of simulations that its current-voltage relation is diverse across interesting wiring configurations. The wiring configuration is mapped to the weights of a neural network, and we study its capacity for pattern recognition by means of simulations for a toy problem, i.e. classification of  $3 \times 5$  pixel images of the digits **1-8** by a NP internet consisting of 8 clusters, each cluster being a disordered system of 4 NPs in between 8 electrodes.

While our focus has been on NP systems so far, any physical system whose input-output relation is diverse across configurations, could realize some form of intelligence. In Chapter 6, we model intelligence as the capacity to relate patterns to data. We map input-output relations of a physical system to a table of datum-pattern relations, from which we calculate an IQ metric. This methodology is applied to experiments on a nanomaterial cluster of dopant atoms. Future experiments to demonstrate the practicality of our IQ metric are outlined.

Finally, in Chapter 7, we reflect upon the impact of our work within the scientific community and note down avenues for further research.

## REFERENCES

- [Shankar2010] Maran, K. (Producer), Shankar, S., Sujatha, R., Vairamuthu, M. K. (Writers), & Shankar, S. (Director). (2010). *Enthiran* [Motion Picture]. India.
- [Kurzweil2010] Kurzweil, R. (2010). *The singularity is near*. Gerald Duckworth & Co.
- [LeCun2015] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436.
- [Schmidhuber2015] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- [Mnih2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529.
- [Silver2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587), 484-489.
- [Nielsen2016] Nielsen, M. (2016). How Google’s AlphaGo Imitates Human Intuition. *Atlantic Monthly*.
- [Jouppi2017] Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., ... & Boyle, R. (2017, June). In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture* (pp. 1-12). ACM.
- [Han2016] Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M. A., & Dally, W. J. (2016, June). EIE: efficient inference engine on compressed deep neural network. In *Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on* (pp. 243-254). IEEE.
- [Chen2017] Chen, Y. H., Krishna, T., Emer, J. S., & Sze, V. (2017). Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits*, 52(1), 127-138.
- [Gomes2017] Gomes, L. (2017). Special report: Can we copy the brain?-The neuromorphic chip’s make-or-break moment. *IEEE Spectrum*, 54(6), 52-57.
- [Merolla2014] Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., ... & Brezzo, B. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197), 668-673.
- [Abbott2016] Abbott, L. F., DePasquale, B., & Memmesheimer, R. M. (2016). Building functional networks of spiking model neurons. *Nature neuroscience*, 19(3), 350.

14. [Eliasmith2012] Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., & Rasmussen, D. (2012). A large-scale model of the functioning brain. *science*, 338(6111), 1202-1205.
15. [Esser2015] Esser, S. K., Appuswamy, R., Merolla, P., Arthur, J. V., & Modha, D. S. (2015). Backpropagation for energy-efficient neuromorphic computing. In *Advances in Neural Information Processing Systems* (pp. 1117-1125).
16. [Lloyd2000] Lloyd, S. (2000). Ultimate physical limits to computation. *Nature*, 406(6799), 1047.
17. [Hopfield1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8), 2554-2558.
18. [Watts1998] Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *nature*, 393(6684), 440.
19. [Wiesenfeld1995] Wiesenfeld, K., & Moss, F. (1995). Stochastic resonance and the benefits of noise: from ice ages to crayfish and SQUIDS. *Nature*, 373(6509), 33.
20. [Toffoli2004] Toffoli, T. (2004). Nothing Makes Sense in Computing Except in the Light of Evolution. *IJUC*, 1(1), 3-29.
21. [Miller2002] Miller, J. F., & Downing, K. (2002). Evolution in materio: Looking beyond the silicon box. In *Evolvable Hardware, 2002. Proceedings. NASA/DoD Conference on* (pp. 167-176). IEEE.
22. [Harding2004] Harding, S., & Miller, J. F. (2004, June). Evolution in materio: A tone discriminator in liquid crystal. In *Evolutionary Computation, 2004. CEC2004. Congress on* (Vol. 2, pp. 1800-1807). IEEE.
23. [Broersma2012] Broersma, H. J., Gomez, F., Miller, J., Petty, M., & Tufte, G. (2012). Nascence project: Nanoscale engineering for novel computation using evolution. *International journal of unconventional computing*, 8(4), 313-317.
24. [Bose2015] Bose, S. K., Lawrence, C. P., Liu, Z., Makarenko, K. S., Van Damme, R. M. J., Broersma, H. J., & Van Der Wiel, W. G. (2015). Evolution of a designless nanoparticle network into reconfigurable boolean logic. *Nature nanotechnology*, 10(12), 1048-1052.
25. [Han2015] Han, J. (2015). Computing: Naturally random. *Nature nanotechnology*, 10(12), 1011.



## 2 Single-electron tunnelling in nanoparticle networks

In the year 1947, it was found in the Netherlands that microscopic granular metal films exhibit extremely nonlinear current-voltage relations at low temperatures and low electric fields. This discovery was explained as a result of the thermal energy and electromotive force being insufficient to provide the electrostatic energy required to separate charges over distant metal grains [Gorter1951]. The phenomenon is now known as Coulomb blockade, and was rigorously explained in 1985 by the ‘orthodox’ theory of single-electron tunnelling (SET) [Averin1986]. A good review of SET physics and circuit designs can be found in [Likharev1999, Wasshuber2001]. To date, applications of SET for computing are limited due to spatially random background charges and variance in device dimensions at the nanoscale. However, we can take a complementary point of view, and regard this diversity as an asset to realize a natural computer, where functionality could emerge from ‘rich physics’ in networks tuned by evolutionary learning. That is what we demonstrated in our work, Evolution of a disordered nanoparticle cluster into reconfigurable Boolean logic [Bose2015]. In this chapter, we shall understand what this rich physics is by means of theory and simulation of the current-voltage relations in SET networks.

### 2.1 SET network model of a NP cluster

Consider a cluster of islands of metallic NPs separated by molecular monolayer coatings, and surrounded by electrodes (as shown in Figure 2.1a). Each island is labelled from  $r = 1$  to  $N$ , and a bias, output ground, gate electrode are labelled by  $s = B, O, G$ , respectively. Note that we avoid the commonly used terms source/drain electrodes because the flow of electrons in SET networks is not always unidirectional. The molecules form tunnel junctions such that the resistance for electron tunnelling from island  $r$  to island  $r'$  is  $R_{rr'} = R_{r'r}$  and the mutual capacitance is  $C_{rr'} = C_{r'r}$ . For tunnelling between island  $r$  and electrode  $s$ , the values are  $R_{sr}$  and  $C_{sr}$ , respectively. The self-capacitance of island  $r$  is  $C_{rr}$ .

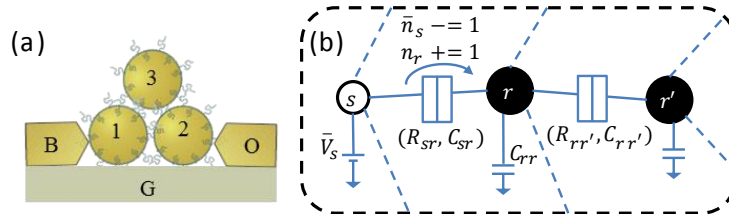


Figure 2.1. (a) A NP cluster. The green hairs are molecules separating NPs from each other and the electrodes. The electrodes are labelled as B for bias, O for output ground and G for gate. (b) Section of a SET network showing an applied voltage  $\bar{V}_s$  on electrode  $s$ , two islands  $r$  and  $r'$ , self-capacitances, and tunnel junctions which are represented by a box with a cut in the middle. The arrow represents tunnelling from electrode  $s$  to island  $r$  across a tunnel junction. Due to that tunnel event, electron number  $\bar{n}_s$  is decreased by 1 and  $n_r$  is increased by 1.

Electrons tunnel through the resulting SET network as shown in Figure 2.1b, affecting the electron number  $n_r$  on an island or  $\bar{n}_s$  on an electrode (the bar is used for parameters of electrodes). To determine the occurrence of these tunnel events, we shall compute the change in free energy for tunnelling across each junction.

### 2.1.1 Free energy of a tunnel event

Given applied voltages  $\bar{V}_s$  and island charges  $q_r = -n_r e$ , the island potentials  $V_r$  are obtained by solving the charge balance equations

$$\begin{aligned} q_r &= C_{rr} V_r + \sum_{r' \neq r} C_{rr'} (V_r - V_{r'}) + \sum_s C_{sr} (V_r - \bar{V}_s) \\ &= \left( \sum_{r'} C_{rr'} + \sum_s C_{sr} \right) V_r - \sum_{r' \neq r} C_{rr'} V_{r'} - \sum_s C_{sr} \bar{V}_s, \end{aligned}$$

Equation 2.1

where we see contributions due to the self-capacitance, mutual capacitances between island  $r$  and all the other islands, and stray capacitances between island  $r$  and all the electrodes. Defining a *background* electron number  $m_r \equiv -\sum_s C_{sr} \bar{V}_s / e$ , the entire system of equations can be cast into a vector and matrix form

$$-ne = \mathbf{C}V + me$$

Equation 2.2

$$\Rightarrow V = -\Phi(m + n),$$

Island potential Equation 2.3

where  $\mathbf{C}$  is the corresponding capacitance matrix (with  $\mathbf{C}_{rr'} = -C_{rr'}$ ,  $\mathbf{C}_{rr} = \sum_{r'} C_{rr'} + \sum_s C_{sr}$ ), and we introduce  $\Phi = e\mathbf{C}^{-1}$  as the charging potential matrix. Note that the largest matrix element of  $\Phi$ , defined as the critical potential  $\varphi$  of the SET network, is analogous to the ionization potential in atoms, and will be used later as a measure for applying voltages in simulations.

The charge induced on an electrode  $s$  is given by  $\bar{q}_s = \sum_r C_{sr} (\bar{V}_s - V_r)$ . The electrostatic potential energy due to all charges is  $U = \frac{1}{2} \sum_r q_r V_r + \frac{1}{2} \sum_s \bar{q}_s \bar{V}_s$ . The work done by the applied voltages on the system is  $W = \sum_s (\bar{q}_s + e\bar{n}_s) \bar{V}_s$ .

Note that

$$\begin{aligned} \sum_s \bar{q}_s \bar{V}_s &= \sum_r \sum_s C_{sr} (\bar{V}_s - V_r) \bar{V}_s = \sum_r \left( -\sum_s C_{sr} \bar{V}_s \right) V_r + \sum_r \sum_s C_{sr} \bar{V}_s^2 \\ &= \sum_r \left( em_r V_r + \sum_s C_{sr} \bar{V}_s^2 \right). \end{aligned}$$

Equation 2.4

The free energy of the system is,



$$\begin{aligned}
F &= U - W = \frac{1}{2} \sum_r q_r V_r - \frac{1}{2} \sum_s \bar{q}_s \bar{V}_s - e \sum_s \bar{n}_s \bar{V}_s \\
&= \frac{1}{2} \sum_r \left( -en_r V_r - em_r V_r - \sum_s C_{sr} \bar{V}_s^2 \right) - e \sum_s \bar{n}_s \bar{V}_s.
\end{aligned}$$

Equation 2.5

It is cast into

$$F(\mathbf{n}, \bar{\mathbf{n}}) = \frac{1}{2} \left( e(\mathbf{n} + \mathbf{m}) \cdot \Phi(\mathbf{m} + \mathbf{n}) - \sum_{s,r} C_{sr} \bar{V}_s^2 \right) - e\bar{\mathbf{n}} \cdot \bar{\mathbf{V}}.$$

Equation 2.6

The change in free energy corresponding to a transition from a state  $\mathbf{n}$  to  $\mathbf{n} + \Delta\mathbf{n}$  by tunnelling of  $\Delta n_r$  electrons to each island  $r$  and  $\Delta \bar{n}_s$  electrons to each electrode  $s$  is hence given by,

$$\begin{aligned}
\Delta F(\mathbf{n}, \Delta\mathbf{n}, \Delta\bar{\mathbf{n}}) &= F(\mathbf{n} + \Delta\mathbf{n}, \bar{\mathbf{n}} + \Delta\bar{\mathbf{n}}) - F(\mathbf{n}, \bar{\mathbf{n}}) = e \left( \Delta\mathbf{n} \cdot \left( \Phi(\mathbf{m} + \mathbf{n}) + \frac{1}{2} \Phi \Delta\mathbf{n} \right) - \Delta\bar{\mathbf{n}} \cdot \bar{\mathbf{V}} \right) \\
&= e \left( \Delta\mathbf{n} \cdot \left( -\mathbf{V} + \frac{1}{2} \Phi \Delta\mathbf{n} \right) - \Delta\bar{\mathbf{n}} \cdot \bar{\mathbf{V}} \right)
\end{aligned}$$

Equation 2.7

The tunnel event of a single electron from location  $\alpha$  to  $\beta$  (via junction  $\alpha|\beta$ ) can be represented by using the Kronecker delta notation as  $\Delta n_r = -\delta_{r\alpha} + \delta_{r\beta}$ , and  $\Delta \bar{n}_s = -\delta_{s\alpha} + \delta_{s\beta}$ . For this case, the change in free energy can now be written as

$$\Delta F = -e \Delta_{\beta}^{\alpha} \cdot \left( \mathbf{V} - \frac{\Phi \Delta_{\beta}^{\alpha}}{2} \right).$$

Equation 2.8

where  $\Delta_{\beta}^{\alpha} \equiv \Delta\mathbf{n} \cup \Delta\bar{\mathbf{n}}$  and  $\mathbf{V} \equiv \mathbf{V} \cup \bar{\mathbf{V}}$ .

If the change in free energy is negative, then a tunnel event through that junction is favourable. Next, we shall calculate the tunnel rate for such an event.

### 2.1.2 Tunnelling rates

A general treatment for computing tunnelling rates involving multiple electrons tunnelling simultaneously across multiple junctions is described in [Averin1992]. For SET networks with high charging energy  $e\varphi \gg k_B T$ , and tunnel resistance  $R \gg h/e^2$ , it is a good approximation to only consider a single electron tunnelling through a junction (island-island or electrode-island) at the rate  $\Gamma = [-\Delta F/e^2 R]$ , where we introduce the *rectifier* function  $[x] \equiv \max(0, x)$ . We shall deal with only such *ideal* SET networks in here. Thus the tunnelling rate through a junction  $\alpha|\beta$  can be written as

$$\Gamma_{\beta}^{\alpha}(n) = \left[ \Delta_{\beta}^{\alpha} \cdot \left( \mathbf{V} - \frac{\Phi \Delta_{\beta}^{\alpha}}{2} \right) / eR_{\alpha\beta} \right]$$

Equation 2.9

Since our tunnel junctions are symmetric with  $R_{\alpha\beta} = R_{\beta\alpha}$  (although others may use asymmetric junctions for functional advantages [Matsumoto1997]), we obtain a convenient expression for the bi-directional tunnelling rate as

$$\Gamma_{\alpha\beta}(\mathbf{n}) \equiv \left( \Gamma_{\beta}^{\alpha}(\mathbf{n}) - \Gamma_{\alpha}^{\beta}(\mathbf{n}) \right) = \left( \Delta_{\beta}^{\alpha} \cdot \mathbf{r} \left| \frac{\Delta_{\beta}^{\alpha} \cdot \Phi \Delta_{\beta}^{\alpha}}{2} \right| \right) / e R_{\alpha\beta},$$

Tunnelling rate Equation 2.10

where we introduce a *thresholding* function  $(x|y) \equiv [x - y] - [-x - y]$ .

Note that for  $(x|y) = 0, y \geq 0$  it follows that  $|x| \leq y$ . Hence, the threshold (stability) condition for  $\Gamma_{\alpha\beta}(\mathbf{n}) = 0$  is that

$$|\Delta_{\beta}^{\alpha} \cdot \mathbf{r}| \leq \left( \frac{\Delta_{\beta}^{\alpha} \cdot \Phi \Delta_{\beta}^{\alpha}}{2} \right).$$

Equation 2.11

Thus we can derive an island-island tunnelling threshold condition for  $\Gamma_{rr'}(\mathbf{n}) = 0$  as

$$|V_r - V_{r'}| \leq \left( \frac{\Delta\Phi_{rr'} + \Delta\Phi_{r'r}}{2} \right),$$

Equation 2.12

where  $\Delta\Phi_{rr'} \equiv (\Phi_{rr} - \Phi_{r'r'})$ , and electrode-island tunnelling threshold condition for  $\Gamma_{sr}(\mathbf{n}) = 0$  as

$$|V_r - \bar{V}_s| \leq \left( \frac{\Phi_{rr}}{2} \right).$$

Equation 2.13

Using these threshold conditions, we shall analytically describe the ‘stability regions’ where tunnelling rates are zero across all junctions.

### 2.1.3 Charge stability diagram

For applied voltages  $\bar{\mathbf{V}}$ , a state  $\mathbf{n}$  is stable when tunnelling across all junctions is classically forbidden because  $\Gamma_{\alpha\beta}(\mathbf{n}) = 0$ . Furthermore, a state is metastable during a time interval  $\tau$  when tunnelling is quantum mechanically improbable because  $\Gamma_{\alpha\beta}(\mathbf{n})\tau \ll 1$ . Thus, based on the threshold condition for  $\Gamma_{\alpha\beta}(\mathbf{n}) = 0$  we can obtain a stability region for  $\bar{\mathbf{V}}$ . For a basic SET network with electrodes  $s \in \{B, O, G\}$  we set  $\bar{\mathbf{V}} = \{\bar{V}_B, 0, \bar{V}_G\}$ . The stability region is then defined with  $(\bar{V}_G, \bar{V}_B)$  as the  $(x, y)$  axes as shown in Figure 2.2.

To simplify the Island potential Equation 2.3, we define an electrode-island coupling factor  $\kappa_{sr} \equiv \frac{\Phi_r \cdot \mathbf{c}_s}{e}$ , so that

$$V_r = -\Phi_r \cdot (\mathbf{m} + \mathbf{n}) = \kappa_{Gr} \bar{V}_G + \kappa_{Br} \bar{V}_B - \Phi_r \cdot \mathbf{n}$$

Equation 2.14

because  $-\Phi_r \cdot \mathbf{m} = \Phi_r \cdot \Sigma_s \frac{\mathbf{c}_s}{e} \bar{V}_s = \Sigma_s \kappa_{sr} \bar{V}_s$ .

For stability across junction  $r|r'$ , we derive from Equation 2.12 that

$$|(\kappa_{Gr} - \kappa_{Gr'})\bar{V}_G + (\kappa_{Br} - \kappa_{Br'})\bar{V}_B - (\Phi_r - \Phi_{r'}) \cdot \mathbf{n}| \leq \left( \frac{\Delta\Phi_{rr'} + \Delta\Phi_{r'r}}{2} \right),$$

Equation 2.15

which is typically a stability band as illustrated in Figure 2.2a.

Similarly, we derive from Equation 2.13 that for stability across junction O|r we require

$$|\kappa_{Gr}\bar{V}_G + \kappa_{Br}\bar{V}_B - \Phi_r \cdot \mathbf{n}| \leq \left( \frac{\Phi_{rr}}{2} \right),$$

Equation 2.16

and for stability across junction B|r we require

$$|\kappa_{Gr}\bar{V}_G + (\kappa_{Br} - 1)\bar{V}_B - \Phi_r \cdot \mathbf{n}| \leq \left( \frac{\Phi_{rr}}{2} \right).$$

Equation 2.17

The intersection of stability bands of the above two tunnel junctions, results typically in a stability diamond as shown in Figure 2.2b.

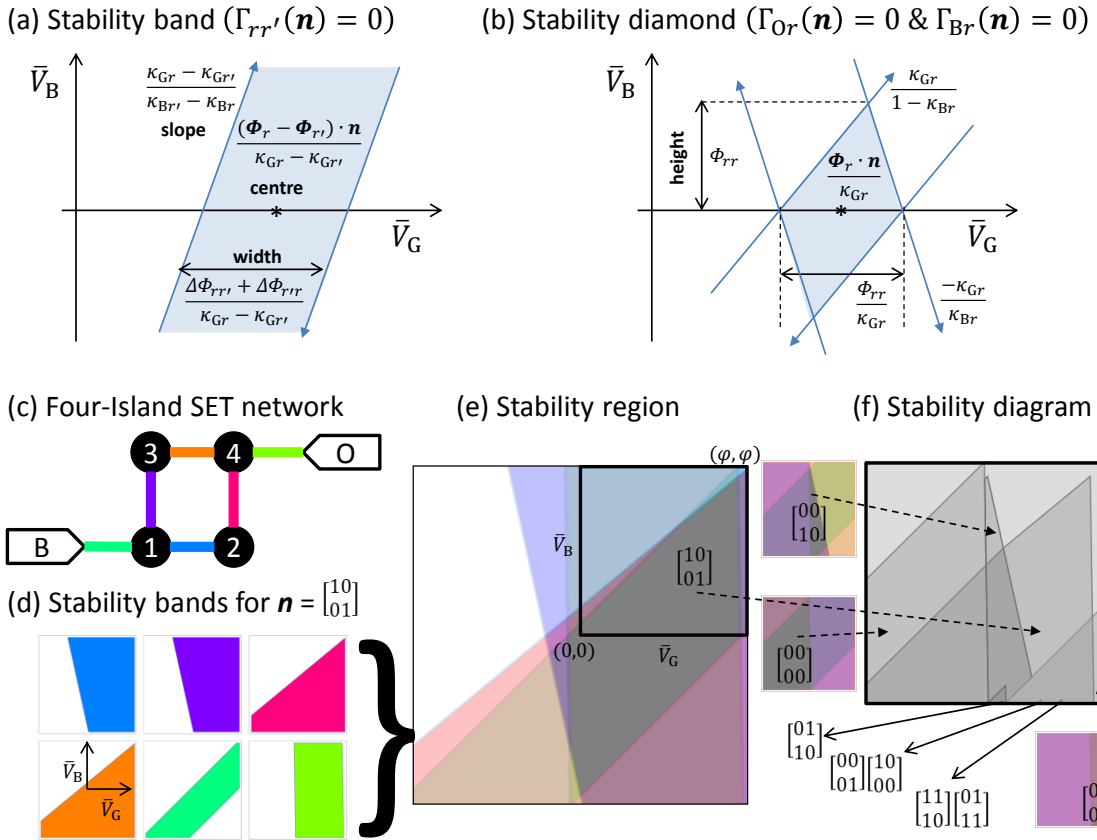


Figure 2.2.(a) Typical stability band due to island-island tunnelling threshold condition, Equation 2.15. (b) Typical stability diamond obtained due to electrode-island tunnelling threshold conditions, Equation 2.16 and Equation 2.17. (c) Canonical four-island network with 6 colour coded tunnel junctions. (d) Stability band (coloured region) for  $\mathbf{n} = \{0,1,1,0\}$  due to each tunnel junction (of the same colour) in the square region defined by  $(\bar{V}_G, \bar{V}_B) = (\pm\varphi, \pm\varphi)$ . (e) Stability region (in grey) as obtained by intersecting the stability bands of all tunnel junctions. (f) Stability diagram obtained by combining stability regions of all possible charge configurations of the above network. Some stability regions are too small to see.

The intersection of the stability bands of all tunnel junctions, results in a convex polygon that defines the complete stability region for a certain charge configuration  $\mathbf{n}$ . By considering stability regions of all possible charge configurations, the stability diagram of an SET network is obtained analytically. In Figure 2.2c-f we perform the above analysis on a four-island uniform SET network introduced in [Shin1999] that is capable of negative differential resistance and hysteretic memory, with  $C_{\alpha\beta} = C$  for the 6 tunnel junctions,  $C_{rr} = 0$  and  $C_{Gr} = 3C$ .

So far in literature [Mizuta2007] stability diamonds, hexagons and octagons have been reported. From our constraints in Equation 2.16 and Equation 2.17 we can reason that for a network with  $M$  tunnel junctions, the stability region is always a convex polygon with at most  $2M$  vertices. The exact positioning and shape of these stability regions can also be expressed analytically in neat matrix-vector expressions that generalize to any SET network. This is a remarkable improvement to the work of Imai [Imai2015a, Imai2015b, Imai2014, Imai2012] focussing on few-particle systems and dealing with ratios of long polynomial functions of capacitances.

Mathematica code to analytically plot stability regions for any SET network is given in Appendix 1. This is an improvement to an online JavaScript simulator by Guenther Lientschnig (<http://lampx.tugraz.at/~hadley/set/setnets/setnets.html>) which calculates stability regions of SET networks with up to 10 islands. Our code may also be used to obtain stability diagrams for SET networks, but since the number of charge configurations scales up exponentially with  $N$ , analytical stability diagrams are impractical for medium scale ( $N > 10$ ) SET networks. We shall now look at charge dynamics after loss of stability.

#### 2.1.4 Charge dynamics at critical potential

To illustrate basic principles of charge dynamics, we consider an SET network of  $N$  islands in series as shown in Figure 2.3, with uniform gate capacitance  $C$  and tunnel junctions between neighbors (of resistance  $R$  and mutual capacitance  $\kappa C$ ). The capacitance matrix is of the form

$$\mathbf{C} = C \begin{pmatrix} 1 + 2\kappa & -\kappa & 0 & \dots & 0 \\ -\kappa & 1 + 2\kappa & -\kappa & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\kappa & 1 + 2\kappa & -\kappa \\ 0 & \dots & 0 & -\kappa & 1 + 2\kappa \end{pmatrix}.$$

When the mutual capacitance is negligible ( $N\kappa \ll 1$ ), we would obtain a simple expression for the charging potential matrix with elements  $\Phi_{rr'} = \varphi \kappa^{|r-r'|}$  with  $\varphi = \frac{e}{C}$ . Such an exponentially falling potential is due to strong screening by nearest neighbours, unlike a weak screening regime as in free space where potentials fall inversely with distance [Whan1996].

When no external voltages are applied ( $\bar{V}_B = \bar{V}_O = 0$ ) and  $\mathbf{m} = \mathbf{0}$  (in the absence of background charges and gate voltage  $\bar{V}_G = 0$ ), the system settles to a charge configuration for which  $\Gamma_{\alpha\beta}(\mathbf{n}) = 0$  across all tunnel junctions. In this case,  $n_r = 0$  is a trivial stable charge configuration. On

setting a sufficiently positive bias  $\bar{V}_B$ , an electron may be pumped through the network by a chain of tunnel events  $\Gamma_B^1(n_r = 0) \rightarrow \Gamma_1^2(n_r = -\delta_{r1}) \rightarrow \dots \Gamma_{x+1}^x(n_r = -\delta_{rx}) \dots \rightarrow \Gamma_N^O(n_r = -\delta_{rN})$ . As shown in Figure 2.3, an electron initially tunnels from island 1 to electrode B, introducing a charge of  $+e$  into the network, which further induces a cascade of tunnelling events from island  $x + 1$  to island  $x$  for  $x = 1$  to  $N - 1$ , and a final tunnel event from electrode O to island  $N$  that resets all charges to 0. This process continues in a cycle and is measured as a ‘hole’ current from electrode B to O. If  $\tau_\beta^\alpha = \frac{1}{\Gamma_\beta^\alpha}$  is the expected duration for a tunnel event, we estimate the current as  $\frac{e}{\tau_B^1 + \tau_1^2 + \dots + \tau_N^O}$ , derived from the total duration for the sequence of all tunnel events.

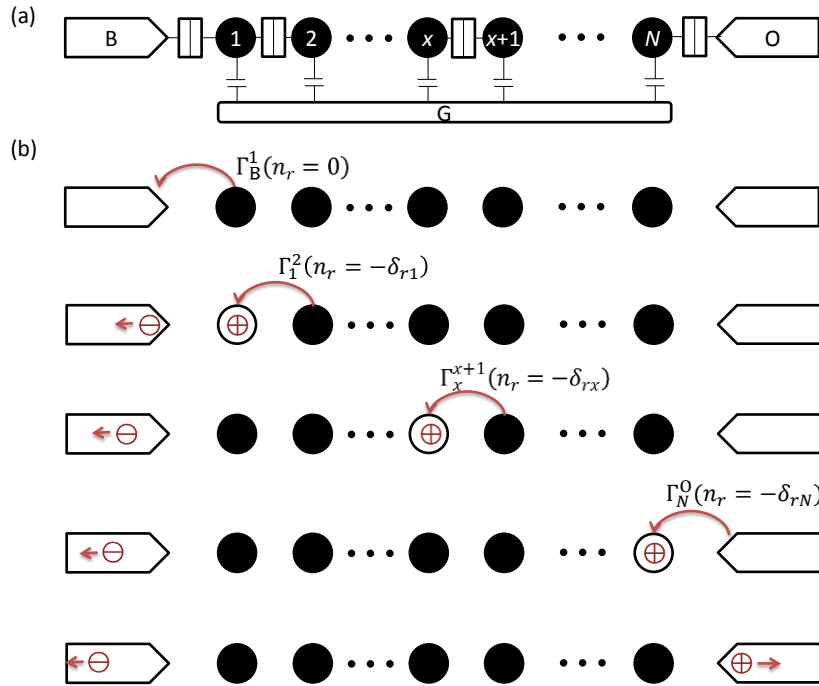


Figure 2.3. (a) Schematic showing the series SET network. (b) Sequence of tunnel events resulting in current. From top to bottom every time one tunnel event takes place.

Note that  $m_r = -\frac{\kappa C \bar{V}_B}{e} \delta_{r1}$  and using Equation 2.9 we can derive that when  $N\kappa \ll 1$  and  $\bar{V}_B = \varphi = \frac{e}{C}$ ,

$$\Gamma_x^{x+1}(n_r = -\delta_{rx}) = \left[ \frac{(\Phi_{x+1} - \Phi_x) \cdot m + \frac{\Phi_{xx} - \Phi_{x+1,x+1}}{2}}{eR} \right] = \left[ \frac{(\Phi_{x,1} - \Phi_{x+1,1}) \kappa C \bar{V}_B}{e^2 R} \right] = \left[ \frac{(1-\kappa) \kappa^x \bar{V}_B}{eR} \right] \approx \frac{\kappa^x}{RC},$$

$$\Gamma_B^1(n_r = 0) = \left[ \frac{\bar{V}_B + \Phi_1 \cdot m - \Phi_{11}/2}{eR} \right] = \left[ \frac{\bar{V}_B \left( 1 - \frac{\Phi_{11} \kappa C}{e} \right) - \Phi_{11}/2}{eR} \right] = \left[ \frac{\bar{V}_B (1-\kappa) - \varphi/2}{eR} \right] \approx \frac{1}{2RC}, \text{ and}$$

$$\Gamma_N^O(n_r = -\delta_{rN}) = \left[ \frac{-\Phi_N \cdot m - \bar{V}_O + \Phi_{NN}/2}{eR} \right] = \left[ \frac{\Phi_{NN}/2 + \Phi_{N1} \frac{\kappa C \bar{V}_B}{e}}{eR} \right] = \left[ \frac{\varphi/2 + \kappa^N \bar{V}_B}{eR} \right] \approx \frac{1}{2RC}.$$

Here we have  $\tau_{1B} = \tau_{ON} = 2RC$  and  $\tau_{x+1,x} = RC\kappa^{-x}$ . This means the current is  $\frac{e\kappa^{N-1}}{RC}$  as determined by the rate limiting step of  $\tau_{N,N-1}$ . The threshold condition for Coulomb blockade,  $\bar{V}_B = \frac{e}{2C(1-\kappa)}$ , is independent of  $N$ .

Except for a limited case of ordered and disordered SET networks [Geigenmüller1989, Middleton1993, Bakhvalov1991, Šuvakov2010], analytical expressions for charge dynamics through SET networks are still unwieldy, and thus we resort to numerical simulations.

## 2.2 Simulation methods

The standard approaches [Wasshuber2001] to simulate charge dynamics keep track of the system's *microstate*  $i$ , which reflects the instantaneous integer number of excess electrons on each island. If we operate within a charging regime of 0 or 1 excess electrons, then each microstate can be represented by a binary string of length  $N$ . For example,  $i = 011$  in a system of three islands at a charge configuration  $\mathbf{n}(i) = \{0,1,1\}$ . There are  $2^N$  such microstates  $i$  over which the system evolves in time by tunnelling events (Markov jump processes). The microstate jumps from  $i$  to  $j$  at a rate  $\Gamma_{ij} = \Sigma_{\alpha,\beta} \Gamma_{\beta}^{\alpha}(\mathbf{n}(i))$  obtained by summing over all possible tunnelling events ( $\alpha, \beta: \mathbf{n}(j) = \mathbf{n}(i) + \Delta_{\beta}^{\alpha}$ ).

For certain applied voltages, the system settles over time into a stable charge configuration  $\mathbf{n}(k)$ , such that  $\Gamma_{kj} = 0$  for all  $j$ . Then the microstate probability can be indicated with help of the Kronecker delta notation:  $P_i = \delta_{ik}$ , where  $k$  is known as an absorbing state.

In other cases, the system may end up cycling among a set of microstates with a stationary distribution  $\{P_i\}$ , resulting in a current flow  $I_{\beta}^{\alpha} = \Sigma_i P_i \Gamma_{\beta}^{\alpha}(\mathbf{n}(i))$ . Here the distribution is derived by solving the conditions of static equilibrium  $\Sigma_{j \neq i} [\Gamma_{ij} P_j - \Gamma_{ji} P_i] = 0$  and total probability  $\Sigma_i P_i = 1$ . For example, in Figure 2.4a, a chain of source, sink and interparticle tunnelling events results in the microstate jumping from 010 to 110 to 100 and back to 100. This can repeat in a cycle to yield a stationary distribution such as  $\{P_{010} = 0.7, P_{110} = 0.2, P_{100} = 0.1 \text{ and } P_{\text{others}} = 0\}$  for  $\Gamma_{110}^{010} = 1, \Gamma_{100}^{110} = 3.5, \Gamma_{010}^{100} = 7$  and  $\Gamma_{\text{others}}^{010,110,100} = 0$ .

In more complex cases [Šuvakov2009], the current flow can fluctuate over time due to a non-stationary distribution  $\{P_i(t)\}$ . For complex SET networks, it is not *a priori* known what the resulting mode of charge dynamics is, and hence for the most accurate simulations we need to solve a master equation (ME) system (see Figure 2.4b) to determine  $\{P_i(t)\}$ . This is extremely time-consuming for large SET networks, and thus Monte Carlo (MC) methods [Wasshuber1997] are employed in practice to generate sufficiently accurate results at a fraction of the cost.

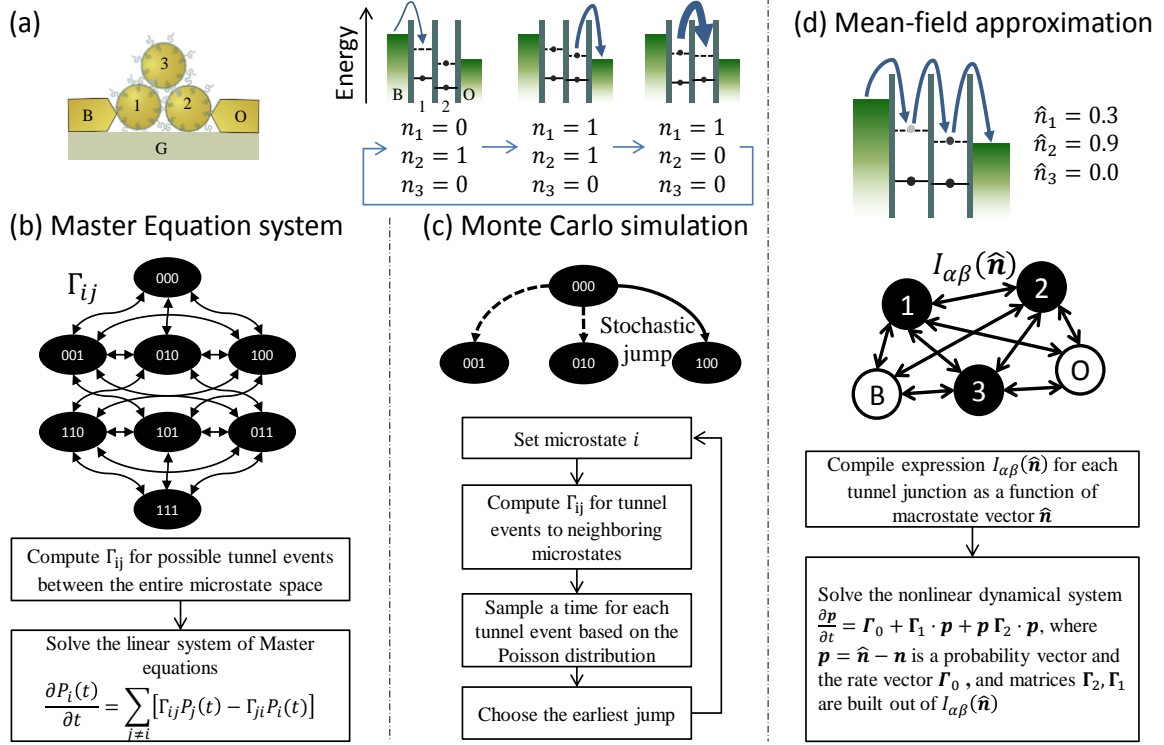


Figure 2.4. (a) An example SET network of 3 islands where applied voltages result in a cycle of tunnel events. The energy level diagrams show the source at an electrochemical potential energy  $= -e\bar{V}_B$ , tunnel junction B|1, electronic occupancy on island 1, tunnel junction 1|2, electronic occupancy on island 2, tunnel junction 2|O, and sink at ground level  $E = 0$ . The arrow thickness represents the tunnelling rate across that junction. There is no tunnelling to island 3 so that  $n_3 = 0$  all the time. (b) The ME method fully describes the time-evolution of microstates by computing all possible transition rates  $\Gamma_{ij}$ . (c) The MC method provides a cheaper solution by iteratively exploring a subspace in vicinity of the relevant microstate. (d) The MF method provides an approximate solution for the macrostate  $\hat{n}$  by balancing the mean currents  $I_{\alpha\beta}(\hat{n})$  in each junction.

### 2.2.1 Monte Carlo method

The MC simulation method is outlined in Figure 2.4c. If we start near an absorbing state  $k$ , then one may expect that only few iterations are necessary to establish stability. On opening up tunnelling pathways by modulation of gate or bias voltages,  $\mathbf{n}(k)$  may lose stability, and there is a need to sample many more tunnel events either to measure a current flow or to settle into a new absorbing state  $k'$ . So in case of large networks with massively parallel tunnelling pathways, the cost of simulating SET networks to a good precision rises significantly. For a network with  $N$  islands, it costs  $\sim N^3$  operations to compute  $\Phi$  and initialize  $\mathbf{V}$ . Thereafter, evaluation of  $\Gamma$  for each of the  $M$  tunnel junctions is required to choose the next tunnel event  $\Delta_{\beta}^{\alpha}$ , and  $\mathbf{V}$  is updated by  $\Delta V_r = \Phi_{r\alpha} - \Phi_{r\beta}$ . Thus it costs  $\sim (M + N)$  operations per tunnel event. To estimate the current  $I_0$ , at least  $fM$  events are required if a fraction  $f$  of tunnel junctions are contributing to the output channel. In addition, non-contributing tunnel junctions carrying stray currents take at least  $g = \frac{e\Gamma_{\max}}{I_0}$  events where  $\Gamma_{\max}$  is the

maximum rate among them. In total, this amounts to a cost of at least  $\sim((fM + g)(M + N) + N^3)$  operations. The worst-case complexity is  $\sim N^4$  for  $f = 1, M = N^2$ .

### 2.2.2 Novel mean-field approximation

The MC method statistically estimates charge dynamics in a physically realistic sense by tracking instantaneous details such as microstates, which may be unnecessary if the goal is to only measure a mean flow of current. Thus we propose a mean-field (MF) approximation (see Figure 2.4d) where a condition for equilibrium current flow is established by solving for the system's *macrostate* (vector  $\hat{\mathbf{n}}$  of dimension  $N$ ), which reflects the average number of excess electrons  $\hat{n}_r$  on each island  $r$ .

We assume that the instantaneous number of electrons on island  $r$  is an integer that is either just lower ( $l = 0$ , floor state) or just higher ( $l = 1$ , ceiling state) than  $\hat{n}_r$ . It is denoted by  $n_{rl} \in \{\lfloor \hat{n}_r \rfloor, \lceil \hat{n}_r \rceil\}$  occurring at a probability  $p_{rl} \in \{1 - p_r, p_r\}$ , respectively. Here  $\hat{n}_r = \sum_l p_{rl} n_{rl} = \lfloor \hat{n}_r \rfloor + p_r$ . When considering tunnel events that involve an island  $\alpha$  at a local state  $l$ , the effective state  $n_r$  is taken such that  $n_{r=\alpha} = n_{\alpha l}$  and  $n_{r \neq \alpha} = \hat{n}_r$ . So we introduce an *effective* operator  $\epsilon_{\alpha l}$  such that  $n_r = \epsilon_{\alpha l} \hat{n}_r = \hat{n}_r + (l - p_\alpha) \delta_{r\alpha}$

The charging rate is then

$$e \frac{\partial \hat{n}_r}{\partial t} = \sum_s I_{sr}(\hat{\mathbf{n}}) + \sum_{r'=1 \text{ to } N} I_{rr'}(\hat{\mathbf{n}}),$$

Equation 2.18

where the currents are the average tunnelling activity over probable local states, given by

$$I_{sr}(\hat{\mathbf{n}}) = e \sum_{l \in \{0,1\}} p_{rl} \Gamma_{sr}(\epsilon_{rl} \hat{\mathbf{n}})$$

Equation 2.19

and

$$I_{r'r}(\hat{\mathbf{n}}) = e \sum_{\substack{l' \in \{0,1\} \\ l \in \{0,1\}}} p_{r'l'} p_{rl} \Gamma_{r'r}(\epsilon_{r'l'} \epsilon_{rl} \hat{\mathbf{n}}).$$

Equation 2.20

The tunnelling rates can be derived from Tunnelling rate Equation 2.10 as

$$\Gamma_{sr}(\epsilon_{rl} \hat{\mathbf{n}}) = \frac{(\Delta V_{sr} - l \Phi_{rr} \left| \frac{\Phi_{rr}}{2} \right|)}{e R_{sr}}$$

Equation 2.21

and

$$\Gamma_{r'r}(\epsilon_{r'l'} \epsilon_{rl} \hat{\mathbf{n}}) = \frac{(\Delta V_{r'r} - l \Delta \Phi_{rr'} + l' \Delta \Phi_{r'r} \left| \frac{\Delta \Phi_{rr'} + \Delta \Phi_{r'r}}{2} \right|)}{e R_{r'r}},$$

Equation 2.22



where  $\Delta V_{sr} \equiv \hat{V}_r + \Phi_{rr} p_r - \bar{V}_s$ ,  $\Delta V_{r'r'} \equiv \hat{V}_r + \Delta \Phi_{r'r'} p_r - \hat{V}_{r'} - \Delta \Phi_{r'r'} p_{r'}$  is the floor state potential field and  $\hat{V} = -\Phi(\mathbf{m} + \hat{\mathbf{n}})$  are mean voltages.

If we consider only electrodes  $s \in \{B, O\}$ , then the charging rate (Equation 2.18) can be expressed as

$$\frac{\partial \hat{\mathbf{n}}}{\partial t} = \frac{1}{e} (I_B(\hat{\mathbf{n}}) + I_O(\hat{\mathbf{n}}) + \Sigma \mathbf{l}(\hat{\mathbf{n}})),$$

Equation 2.23

where  $\Sigma$  is the sum over an array's first dimension.  $\Sigma \mathbf{l}(\hat{\mathbf{n}}) = 0$  since  $I_{r'r'}(\hat{\mathbf{n}}) = -I_{r'r'}(\hat{\mathbf{n}})$  and we define  $I_B = \Sigma I_B(\hat{\mathbf{n}})$  and  $I_O = \Sigma I_O(\hat{\mathbf{n}})$ . At dynamic equilibrium, we have  $\Sigma \frac{\partial \hat{\mathbf{n}}}{\partial t} = 0$ , and thus have  $I_O = -I_B$  as the current flow from the network to the ground.

While the nonlinear dynamical system defined above has potential for limit cycles and strange attractors [Strogatz2014], we restrict ourselves to solve for convergence of  $\hat{\mathbf{n}}$  to a stable fixed point  $\mathbf{n}$ . This equilibrium point is either a static charge configuration  $\mathbf{n} = \mathbf{n}(k)$  when all the ‘ $p\Gamma$ ’ terms equal 0, or a dynamic equilibrium with non-zero current flow in the network. The numerical method used to solve for equilibrium points of Equation 2.23 is described in Appendix 2. Note that convergence can be expected only for stable equilibrium points when using our method.

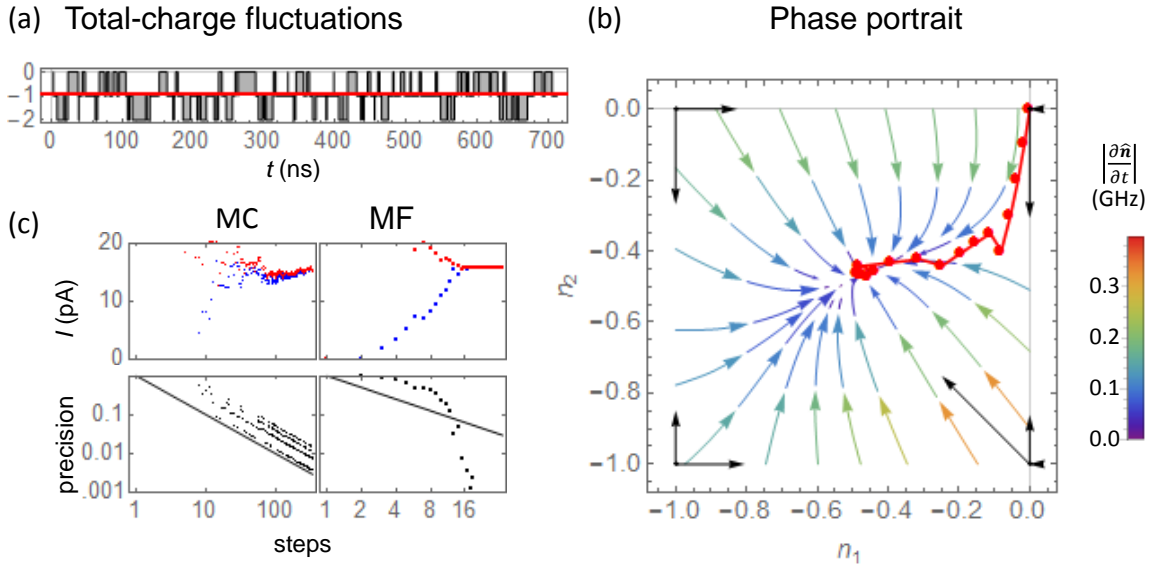


Figure 2.5. Charge dynamics in a disordered SET network of dimension  $N=2$ . (a) Plot of total-charge  $\sum_r n_r(t)$ , as simulated by the MC method. The red baseline is the total-charge at equilibrium  $\sum_r n_r$ , computed by our MF method. (b) Colored streamlines represent the charging rate vector  $\frac{\partial \hat{\mathbf{n}}}{\partial t}$ . Black vectors, each at charge configuration  $n(i)$  pointing towards  $n(j)$ , represent the scaled transition rate  $\frac{e\Gamma_{ij}}{10I_0}$  where  $I_0$  is the equilibrium current. The red trajectory shows steps taken by our MF method to reach equilibrium from origin. (c) Convergence plots for the MF and MC methods, showing current flow ( $I_0$  in blue,  $-I_B$  in red) and precision. Clearly, the MF method converges faster than the MC method (where linear convergence is indicated by the black line with slope = -1).

An illustration of the steps taken by our numerical method to reach equilibrium for a disordered 2-island system (as in Appendix 4) and convergence plot of the MF and MC methods is shown in Figure 2.5. For this example, the MF method converges in fewer than 20 steps whereas the MC method takes more than 100 steps to reach 1% *precision*, which is the relative error (as defined in Appendix 3) between  $I_0$  and  $-I_B$ . For a network with  $N$  islands, it costs  $\sim N^3$  operations to compute  $\Phi$ . Thereafter, evaluation of  $\Gamma$  for each of the  $M$  tunnel junctions is required to calculate  $\frac{\partial \hat{n}}{\partial t}$ , update  $\hat{n}$ , and recompute  $\hat{V}$ . Thus it costs  $\sim (M + N + N^2)$  operations per update step. The number of steps it takes to converge is at least  $N^\alpha$  ( $\alpha = 1$  or  $0.5$  for linear or quadratic convergence). In total, this amounts to a cost of at least  $\sim (N^\alpha (M + N^2) + N^3)$  operations. The worst-case complexity is  $\sim N^3$  for  $\alpha = 1$ ,  $M = N^2$ . Remember that for the MC method the worst-case complexity is  $\sim N^4$ .

### 2.2.3 Comparison of Monte Carlo and mean-field simulations

A netlist specifying RC parameters of a disordered SET network and suitable operating voltages is computer generated (see Appendix 4), and simulations are run on a modern PC using this netlist. The netlist generator and MF method were programmed in Mathematica, and we used a commercial SET simulator for the MC method (SIMON [Wasshuber1997]).

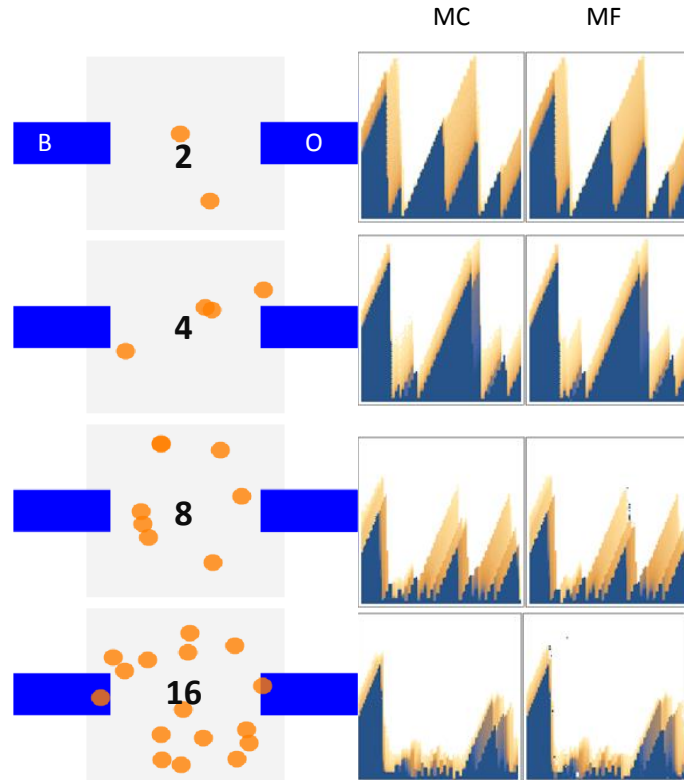


Figure. 2.6. Simulation of disordered SET networks for different numbers of NPs (in orange) between bias and ground electrodes (blue rectangles) on top of a gate dielectric (grey area). The simulated stability diagrams show density plots of current  $I_0$  (colored in log scale as dark blue for fA to white for pA) as a function of  $\bar{V}_B$  (0 V to  $\varphi$  in y-axis) and  $\bar{V}_G$  (0 V to  $2.5 \varphi$  in x-axis).

We see that the stability diagrams in Figure. 2.6 are in good overall agreement for disordered SET networks of up to 16 NPs. In Figure 2.7, we compare the output currents at their critical potential for an exponentially increasing number of NPs. We see that for at least up to  $N = 512$ , the MF method appears to describe the output current accurately in comparison to the MC method. We calculated *accuracy* as the relative error in current  $I_0$  between the two methods. Beyond  $N = 512$ , MC simulations became impractical in runtime for our PC, whereas we could go up to  $N = 4000$  with MF simulations.

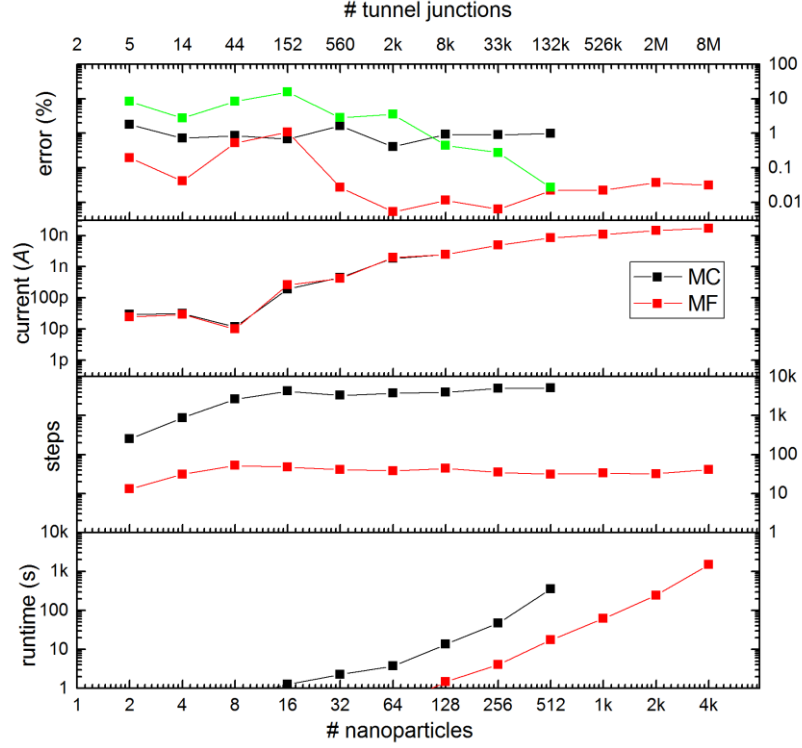


Figure 2.7. Simulation trends for SET networks operating at  $\bar{V}_B = \varphi$  and  $\bar{V}_G = 0$  V. In the top panel, we plot errors in terms of precision for each method (in black for MC, red for MF) and accuracy between the methods (in green). For large networks ( $N > 16$ ), the MF approach is superior to the MC method in terms of rate of convergence when considering the precision and steps taken (or equivalently runtime).

## 2.3 Illustrative simulations

In this section, we simulate different kinds of SET networks with special properties that are relevant for computing. We shall also use these SET networks to elucidate the functional accuracy of the MF approximation in comparison to the MC method.

### 2.3.1 Negative differential resistance (NDR)

In Figure 2.8, we see that even the simplest SET network is capable of NDR. Devices with NDR are sought after by circuit designers, as they can be loaded in series with a small or large resistor (relative to the NDR) and realize a small-signal voltage amplifier or a bistable latch, respectively.

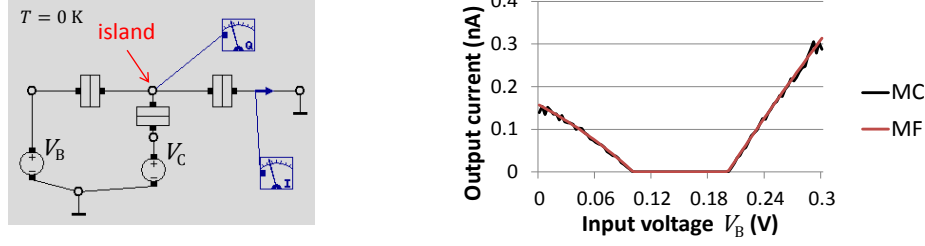


Figure 2.8. An island surrounded by 3 electrodes. The tunnel junctions have  $R = 100 \text{ M}\Omega$  and  $C = 0.16 \text{ aF}$ . On setting the control voltage  $V_C = 300 \text{ mV}$ , and varying input voltage  $V_B$  from 0 to 300 mV we measure an output current to ground as plotted for both simulation methods.

### 2.3.2 Bistability

Bistability is a form of memory that enables sequential logic. In Figure 2.9, we show a SET network inspired from [Wasshuber2001] that has bistability. In the MC simulation, we observe that applying a positive voltage spike ‘sets’ the charge configuration to (1,-1,0) until a negative voltage spike ‘resets’ it to before. A similar quality is observed in the MF simulation where the macrostate converges to either (1,0,0) or (1,-1,0) depending on the initial conditions.

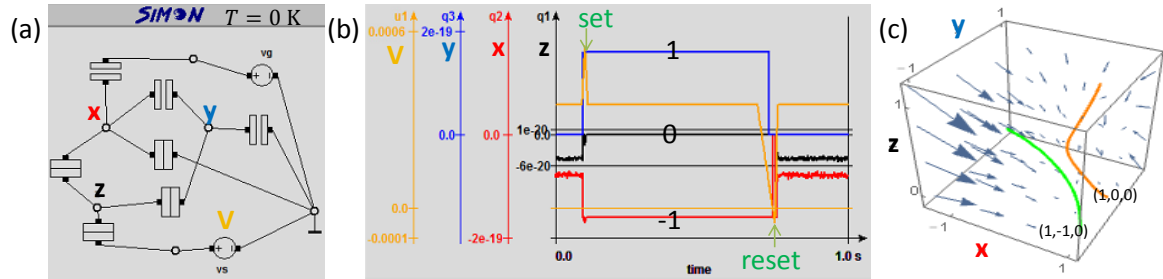


Figure 2.9.(a) Bistable circuit, see Appendix 6 for netlist. (b) MC simulation, applied voltage and charge configurations over time.(c) MF simulation, phase portrait of the resulting dynamical system is sensitive to the initial macrostate and converges to different equilibrium points.

### 2.3.3 Irregularity

Irregularity is not always a virtue, unless we are dealing with capacitances in SET networks. We note that stability diagrams appear to be richer in diversity of current-voltage relations when the capacitances are not multiples of each other as shown in Figure 2.10. This is because periodicity in stability diagrams would be the least common multiple of the periodicities of stability bands in each tunnel junction. Note that if the capacitances are irrational, then the stability diagrams would never be truly periodic.

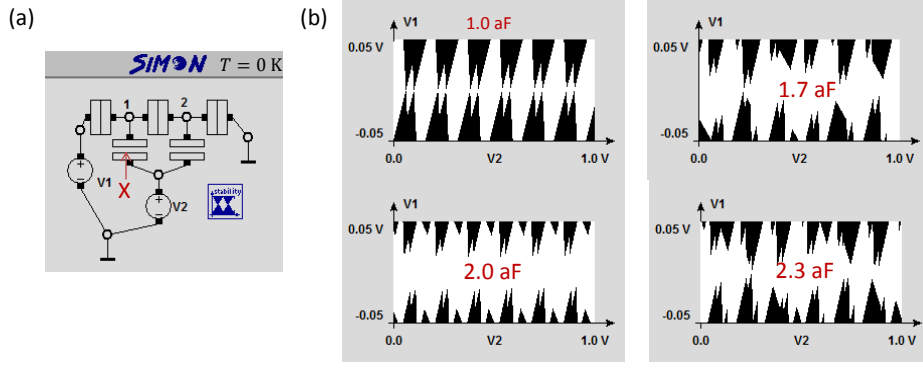


Figure 2.10. (a) A SET network with 2 islands in series. Capacitances are all 1 aF, except for X. (b) Stability diagrams for different values of X.

### 2.3.4 Boolean logic in a regular SET network

In the previous section, we dealt with charge flow in a disordered network between 2 electrodes. Here we look at regular networks between several electrodes as shown in Figure 2.11. We presented MC simulation results in [vanDamme2016] to produce 2-input Boolean logic in the output current by optimizing control voltages using a genetic algorithm. Our MF method reproduces the logic in 3 out of the 4 gates tested.

Here, the MF method is functionally inaccurate for voltage configuration of the OR gate with one of the inputs as high, by converging to a fake-stable macrostate, which blocks tunnelling between the output electrode and the output island. The macrostate is fake-stable because in reality the Coulomb blockade is broken by a significant potential fluctuation on the output island because of current flow through an adjacent island (due to a relatively large interaction potential  $\Phi_{rr'}$ ).

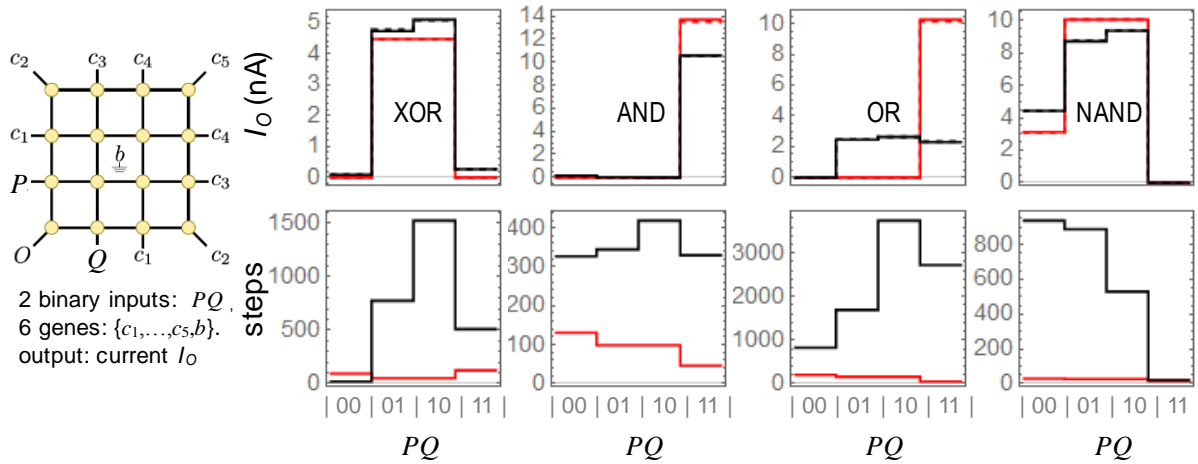


Figure 2.11. Simulation of logic gates found in a 4x4 NP grid [vanDamme2016]. We compare MF (red) and MC (black) results in terms of the current flow and the number of steps taken for a precision of 1%. Refer to Appendix 5 for the actual voltages applied.

### 2.3.5 Multiscale networks

So far we have shown SET networks where islands are nanoparticles with similar order of capacitance values. However, in reality we would also encounter SET networks where nanoparticles are interconnected by conductors of much larger dimensions. In such cases, the capacitances of interconnect islands are orders of magnitude larger than the particle islands.

In Figure 2.12, we look at how the NDR functionality is affected upon introducing an interconnect with self-capacitance  $C_1$ . From the stability diagram, we see that NDR appears to be preserved for most values of  $C_1$  between 0 to 100 aF. In fact, the NDR is observed even for  $C_1 = 1000$  aF after making sure more than sufficient number of tunnel events ( $=10000$ ) were considered to fully charge the interconnect islands. The width of the NDR region (in secondary Coulomb blockade) shrinks a bit though.

Interestingly, a kind of connectionistic NDR or what we shall later define in Chapter 5 Section 3.2 as ‘inhibition’ is seen. For example, increasing  $C_1$  from 5 aF to 10 aF can lead to a drop in current if input voltage is within the secondary Coulomb blockade region.

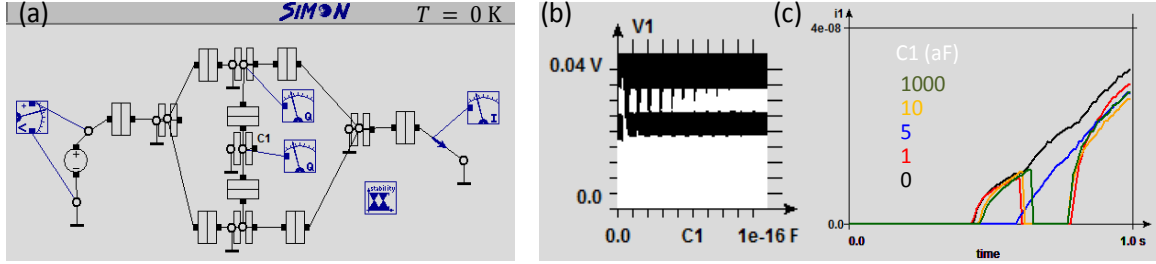


Figure 2.12. (a) The four-island network interconnected by a node of self-capacitance  $C_1$ . Check Appendix 7 for netlist. (b) Stability diagram across  $C_1$ , from 0 to 100 aF. (c) Current in time when voltage is swept from 0 to 40 mV for selected cases of  $C_1$ .

## 2.4 Conclusion

We have shown that our MF method is 10-100 times faster than an MC method for precise current-voltage simulations of disordered SET networks. This makes simulations of large-scale SET networks of 100 NPs ( $10^4$  junctions) feasible in under 1s on a modern PC. Thus the next step would be to simulate large-scale SET networks (now more efficiently by our MF method) and apply them for natural computing.

Our MF method can be proven to be exact for simulating (at  $T = 0$  K) charge-stability diagrams of non-interacting ( $\Phi_{rr'} = 0$ ) SET networks, which are equivalent to circuits of single-electron transistors. Exact simulations of single-electron transistors, for any temperature, are already known to be efficiently implemented in SPICE [Lientschnig2003]. Unfortunately, interacting SET

networks are incompatible with SPICE because it requires independent models for its circuit elements. Thus, our MF method is the only known alternative that is better than a MC method.

We saw that our MF method is capable of reproducing 3 out of the 4 logic gates found by an evolutionary algorithm on MC simulations. Further research needs to be done on the reasons for failure of our MF approach in select cases, and in identifying conditions under which our MF approximation is reliable. To this end, we could learn from mean-field convergence results in other systems of interacting objects such as transport and communication networks [LeBoudec2007].

Apart from comparing the current levels, checking for equivalence between the microstates/macrosate of the MC/MF methods offers more insight into the physical validity of our model. Moreover, solving the dynamical system by a time-integration method opens the door to simulation of non-equilibrium dynamics as well. A systematic design for emergence [Dogaru2008] is recommended by mapping a SET network's physical parameters to the feedback, feedforward and bias interactions of an equivalent (cellular) neural network, as previously investigated in a preliminary version of our MF model by Douwe de Bruijn [Bruijn2015].

## REFERENCES

- [Gorter1951] Gorter, C. J. (1951). A possible explanation of the increase of the electrical resistance of thin metal films at low temperatures and small field strengths. *Physica* 17.8: 777-780.
- [Averin1986] Averin, D. V., & Likharev, K. K. (1986). Coulomb blockade of single-electron tunneling, and coherent oscillations in small tunnel junctions. *Journal of low temperature physics*, 62(3-4), 345-373.
- [Likharev1999] Likharev, K. K. (1999). Single-electron devices and their applications. *Proceedings of the IEEE*, 87(4), 606-632.
- [Wasshuber2001] Wasshuber, C. (2001). *Computational Single-Electronics*. Springer New York.
- [Bose2015] Bose, S. K., Lawrence, C. P., Liu, Z., Makarenko, K. S., van Damme, R. M., Broersma, H. J., & van der Wiel, W. G. (2015). Evolution of a designless nanoparticle network into reconfigurable boolean logic. *Nature nanotechnology*, 10(12), 1048.
- [Averin1992] Averin, D. V., & Nazarov, Y. V. (1992). Macroscopic quantum tunneling of charge and cotunneling. In *Single Charge Tunneling* (pp. 217-247). Springer US.
- [Matsumoto1997] Matsumoto, Y., Hanajiri, T., Toyabe, T., & Sugano, T. (1997). Advantages of the asymmetric tunnel barrier for high-density integration of single electron devices. *Japanese journal of applied physics*, 36(6S), 4143.
- [Shin1999] Shin, M., Lee, S., Park, K. W., & Lee, E. H. (1999). Secondary Coulomb blockade gap in a four-island tunnel-junction array. *Physical Review B*, 59(4), 3160.
- [Mizuta2007] Mizuta, A., Moriya, M., Usami, K., Kobayashi, T., Shimada, H., & Mizugaki, Y. (2007). Coulomb blockade conditions for detailed model of single-electron turnstile device including finite self-capacitances of island electrodes. *Japanese journal of applied physics*, 46(5R), 3144.
- [Imai2015a] Imai, S., & Iwasa, N. (2015). Stability diagrams and turnstile operations of single-common-gate triple-dot single-electron devices with outer junction capacitances different from inner ones. *Japanese Journal of Applied Physics*, 54(6), 064001.
- [Imai2015b] Imai, S., Nakajima, A., & Kobata, T. (2015). Single-electron pumping in single-common-gate triple-dot devices with asymmetric gate capacitances. *Japanese Journal of Applied Physics*, 54(10), 104001.
- [Imai2014] Imai, S., & Moriguchi, S. I. (2014). Single-common-gate triple-dot single-electron devices with side gate capacitances larger than the central one. *Japanese Journal of Applied Physics*, 53(9), 094002.
- [Imai2012] Imai, S., Kato, H., & Hiraoka, Y. (2012). Stability Diagrams of Single-Common-Gate Double-Dot Single-Electron Transistors with Arbitrary Junction and Gate Capacitances. *Japanese Journal of Applied Physics*, 51(12R), 124301.
- [Whan1996] Whan, C. B., White, J., & Orlando, T. P. (1996). Full capacitance matrix of coupled quantum dot arrays: Static and dynamical effects. *Applied physics letters*, 68(21), 2996-2998.
- [Geigenmüller1989] Geigenmüller, U., & Schön, G. (1989). Single-electron effects in arrays of normal tunnel junctions. *EPL (Europhysics Letters)*, 10(8), 765.
- [Middleton1993] Middleton, A. A., & Wingreen, N. S. (1993). Collective transport in arrays of small metallic dots. *Physical review letters*, 71(19), 3198.

17. [Bakhvalov1991] Bakhvalov, N. S., Kazacha, G. S., Likharev, K. K., & Serdyukova, S. I. (1991). Statics and dynamics of single-electron solitons in two-dimensional arrays of ultrasmall tunnel junctions. *Physica B: Condensed Matter*, 173(3), 319-328.
18. [Šuvakov2010] Šuvakov, M., & Tadić, B. (2010). Modeling collective charge transport in nanoparticle assemblies. *Journal of Physics: Condensed Matter*, 22(16), 163201.
19. [Šuvakov2009] Šuvakov, M., & Tadić, B. (2009). Collective charge fluctuations in single-electron processes on nanonetworks. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(02), P02015.
20. [Wasshuber1997] Wasshuber, C., Kosina, H., & Selberherr, S. (1997). SIMON-A simulator for single-electron tunnel devices and circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(9), 937-944.
21. [Strogatz2014] Strogatz, S. H. (2014). Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. Westview press.
22. [vanDamme2016] van Damme, R., Broersma, H., Mikhali, J., Lawrence, C., & van der Wiel, W. (2016, July). A simulation tool for evolving functionalities in disordered nanoparticle networks. In *Evolutionary Computation (CEC), 2016 IEEE Congress on* (pp. 5238-5245). IEEE.
23. [Lientschnig2003] Lientschnig, G., Weymann, I., & Hadley, P. (2003). Simulating hybrid circuits of single-electron transistors and field-effect transistors. *Japanese journal of applied physics*, 42(10R), 6467.
24. [LeBoudec2007] Le Boudec, J. Y., McDonald, D., & Mundinger, J. (2007, September). A generic mean field convergence result for systems of interacting objects. In *Quantitative Evaluation of Systems, 2007. QEST 2007. Fourth International Conference on the* (pp. 3-18). IEEE.
25. [Dogaru2008] Dogaru, R. (2008). Systematic design for emergence in cellular nonlinear networks. *Studies in Computational Intelligence Springer*, 95.
26. [Bruijn2015] Bruijn, D. S. (2015). *Mapping electron tunnelling in a nanoparticle network to a cellular neural network* (Bachelor's thesis). Enschede: University of Twente.



### 3 Fabrication of NP clusters

After the oxygen we breathe, silicon is the most abundant element found in the Earth's crust. And due to its special properties as a semiconductor, silicon literally serves as the bedrock for modern day electronics. However, in our work, we use silicon not for its semiconductor properties. A wafer of heavily doped polysilicon (0.5 mm thick) covered by an insulating layer of thermally grown amorphous silicon dioxide (35 nm thick) serves as a substrate to deposit metallic electrodes up on. For our experiments, a gate voltage is applied to the conductive polysilicon and the oxide layer thickness determines the gate capacitance. For making electrodes, we use the precious metal gold. It is inert, an excellent conductor and of very high density. However due to the inertness, thin films of gold (30 nm) do not adhere well to the oxide layer and thus an adhesion layer of titanium (5 nm) is deposited first. The titanium binds on both sides by forming an oxide with silicon dioxide and an alloy with gold.

#### 3.1 Patterning electrodes on a chip

Patterning electrodes on a chip (of silicon substrate) is similar to printing photographs by the exposure-development method. Here, instead of a photographic film, we use a polymer resist whose chains upon exposure to different wavelengths and intensity can break or link (thus the resist type is called positive or negative). The exposure is defined either simultaneously as in photolithography by shining a flood of light through a mask, or serially as in electron-beam lithography (EBL) by writing with a beam of electrons. In Figure 3.1, we show a process flow to fabricate electrodes on a chip using EBL. First, we put a drop of resist (here acrylic) on the substrate and "spin coat" it. Second, we "bake" the resist so that it hardens and forms a flat layer of about 100 nm thickness. Third, we "expose" the resist to a pattern of electron beams. Fourth, we use a mild solvent to "develop" so that the exposed areas dissolve away because the resist type is positive. Fifth, we "deposit" metal by electron-beam physical vapour deposition (EBPVD). Finally, we use a strong solvent and ultrasonication to dissolve away the remaining resist and "lift-off" the unnecessary metal. In the end, we are left with the desired pattern of metallic electrodes on the substrate.

Usually nanostructures for multiple devices are fabricated by EBL (10 nm resolution) and if the yield is good then microstructures are fabricated by photolithography (2  $\mu\text{m}$  resolution). The alignment of the photolithography pattern to the EBL pattern is crucial, and there has to be sufficient overlap to stitch the patterns into continuous electrodes. So the EBL patterns have features as large as 10  $\mu\text{m}$ . The other way of doing photolithography first may allow for a custom alignment of EBL patterning for each group of electrodes and thus reduce the total area of EBL patterning, which means a savings in patterning time. However, depositing thin nanostructures on top of thick microstructures causes extreme strain at the overlapping junctions. As a recommendation for the future, we suggest

combining photo and e-beam lithography to use the same resist [Carbaugh2017] and a single step of deposition is carried out after both exposures are developed.

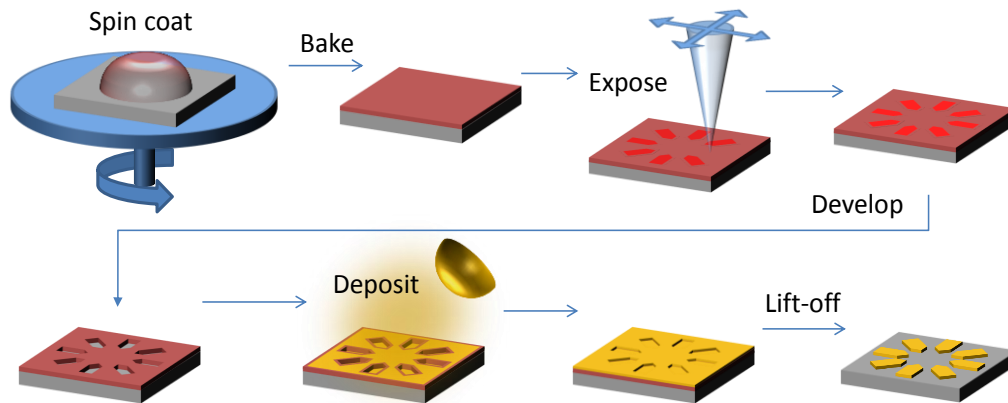


Figure 3.1. Process flow to fabricate electrodes on a silicon substrate. A thin film of acrylic is spin coated and exposed by electron-beam lithography to form bright red patterns; the exposed areas are dissolved away and gold vapour is deposited to leave a pattern of metallic electrodes after the resist is lift-off.

Nanofabrication as described above, requires extremely controlled environments. To put things into perspective, a particle of diameter 1 nm placed on a 12 inch silicon wafer is relatable to a standard table tennis ball placed on earth ( $\frac{10^{-9}}{0.3} \approx \frac{.04}{1.3 \times 10^7}$ ). Similarly, a dust particle of size 5  $\mu\text{m}$  is relatable to a cloud the size of a sports field ( $.04 \times 5000 = 200 \text{ m}$ ). So patterning nanostructures in ambient air is like playing table tennis outdoors and hoping that rain does not spoil your game. Definitely not recommended (especially in the Netherlands) and that is why we work in a cleanroom where ambient air is filtered to reduce micro-clouds of dust by at least 1000 times. Apart from this, the water content in air is kept under control by maintaining the cleanroom around a temperature of 20°C and 50% relative humidity. Inside the machines for EBPVD and EBL, maintaining near-vacuum is necessary. To achieve accurate aiming of the electron beam for defining patterns down to 10 nm resolution, the entire building is vibration controlled to VC-G standards i.e. vibration speed < 0.78  $\mu\text{m/s}$ , which is relatable as above to playing table tennis during cyclonic wind speeds of up to 30 m/s! Using the above described special treatments and more details in Appendix 8 and [Nielen2015], my research colleagues fabricated chips (usually a square of size 1 cm) with multiple groups of electrodes as seen in Figure 3.2.

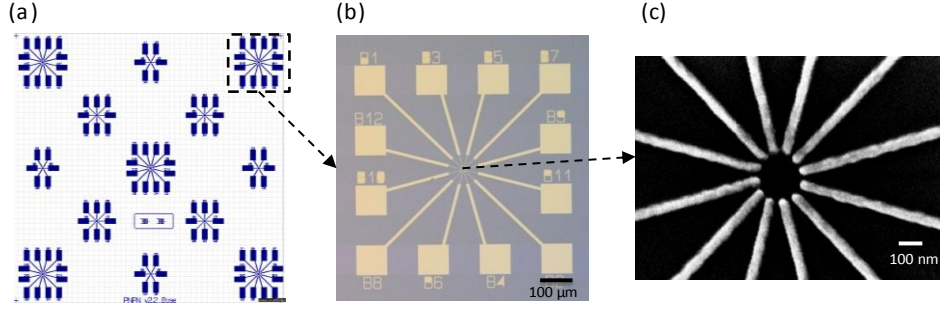


Figure 3.2. (a) A photolithography mask design showing multiple groups of electrodes to pattern on a chip, (b) an optical image showing a group of fabricated electrodes, and (c) an SEM image showing ends of the same electrodes at the nanoscale to form a central gap.

## 3.2 NP trapping by self-assembly

After the chip is ready, we need to fill the central gap between a group of electrodes with nanoparticles to make our device. One could drop cast nanoparticles all over the chip, and hope that nanoparticles also fill in the central gap. For example, in [Andres1996] a planar network of gold nanoparticles (3.7 nm diameter) linked with 2.2 nm long organic molecules was drop casted to fill the 450 nm gap between two thin (30 to 40 nm) gold electrodes. The charging energy was estimated to be  $\sim 100$  meV by computing the capacitance numerically and also by fitting parameters to temperature-dependent conductance measurements ( $G_0 = G_\infty e^{-E_A/k_B T}$ ). However, Coulomb blockade was not observed above 85 K. This is because Coulomb blockading a large number of parallel paths between the electrodes becomes improbable in the presence of spatially random background charges. Keeping this limitation in mind, it is prudent to make paths for charge transport only within the central gap. For that purpose, the electrodes could be coated with an anti-sticking layer so that drop-casted nanoparticles are only in contact near the central gap. Or one could drop nanoparticles exclusively in the central gap by dip-pen nanolithography [Piner1999]. In our work, we choose to go for a proven method [Bernard2007,Pethig2010] that traps nanoparticles between nanogaps based on dielectrophoresis, a physical phenomenon where force is exerted on dielectric particles in a non-uniform and oscillating electric field as depicted in Figure 3.3a-c.

### 3.2.1 Dielectrophoresis

For a spherical particle of density  $\rho_p$  in a medium with an electric field  $\mathbf{E} \sin \omega t$ , the acceleration due to dielectrophoresis (DEP) is given by

$$\langle a_{\text{DEP}} \rangle = \frac{3\epsilon_0}{2\rho_p} p(\omega) \nabla |E_{\text{rms}}|^2$$

Equation 3.1

where the polarization factor  $p(\omega) = \epsilon_M \text{Re} \left[ \frac{\epsilon_P^* - \epsilon_M^*}{\epsilon_P^* + 2\epsilon_M^*} \right]$ , the complex permittivity  $\epsilon^* = \epsilon_0 \epsilon + \frac{i\sigma}{\omega}$  is obtained from the dielectric constant  $\epsilon$  with conductivity  $\sigma$ . Subscripts P and M refer to particle and medium, respectively. The particle is attracted to the electrodes due to positive DEP, as depicted in Figure 3.3d when  $p(\omega) > 0$ . For a bare-gold particle in a non-conducting medium ( $\epsilon_P = 1, \sigma_M \approx 0, \epsilon_M \gg 1$ ), we have  $p(0) \approx \epsilon_M$ ,  $p(\infty) \approx \frac{-\epsilon_M}{2}$  and  $p\left(\omega_c = \frac{\sigma_P}{\sqrt{2}\epsilon_0\epsilon_M}\right) \approx 0$ . With bulk conductivity  $\sigma_P = 4.5 * 10^7$  S/m, the cross-over frequency  $\omega_c$  would be in exahertz, so the polarization factor would always be equal to  $\epsilon_M$  in practice. However, quantitatively gold has a lower conductivity at the nanoscale, and qualitatively they are usually coated with a molecular monolayer L. For a layer thickness  $t_L \ll r_P$  the radius, treating the particle as a core-shell model, the combined permittivity may be derived as  $\epsilon_P^* = \left( \frac{\epsilon_L r_P (\sigma_P + i\omega \epsilon_P)}{t_L \sigma_P + i\omega (\epsilon_P t_L + \epsilon_L r_P)} \right)$  [Li2008]. As seen from Figure 3.3e, the polarization of  $r_P = 10$  nm particles is near maximum at about 1 MHz for our standard choice of materials. The DEP force is weaker for 1 nm particles and requires that  $\epsilon_L/\epsilon_M \approx 1$  for positive DEP.

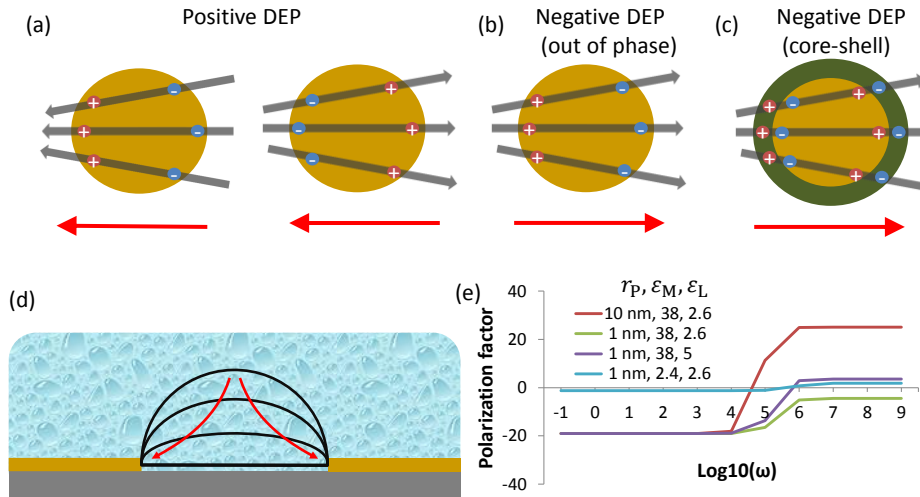


Figure 3.3. Dielectrophoresis. (a) Charges are polarized in phase with an oscillating electric field. Here positive DEP pushes the particle towards the region (red arrow) of increasing field strength as shown for both cases. (b) Charges are polarized out of phase, here we have negative DEP that pushes along a direction of decreasing field strength. (c) Charges are polarized in phase within the core, opposite charges induced at the shell. Here again, we have negative DEP. (d) A cross-sectional view of the trapping region showing electric field lines between electrodes and positive DEP. (e) Plot of  $p(\omega)$  for different particle size, solvent medium (ethylene glycol  $\epsilon_M = 38$  or toluene  $\epsilon_M = 2.4$ ) and molecular layer ( $t_L = 0.1$  nm, octanethiol  $\epsilon_L = 2.6$  or  $\epsilon_L = 5$ ).

Positive DEP alone does not guarantee NP trapping, as it should overcome other forces in the medium. In [Castellanos2003], electro-osmosis and Brownian motion due to collision of solvent molecules against the particle were found to be relevant. Electro-osmosis in water was found to make NP trapping impossible at frequencies below 1 kHz. Here, we shall analytically study the stability and speed of NP trapping by considering Brownian motion.

Let us define the hemispherical region on the central gap (of radius  $l$ ) between electrodes as the trapping region, where positive DEP is in action as depicted in Figure 3.3d. Keeping in mind that the path integral of electric field lines has to be equal to the voltage applied between the electrodes, we approximate the electric field strength at a height  $h$  above the centre as  $E = \frac{V}{2l+h}$ . Putting  $p(\omega) = 1$  and  $\nabla|E_{\text{rms}}|^2 = \frac{V^2}{4l^3}$  in Equation 3.1), we get the maximum acceleration due to DEP as  $a_{\text{DEP}} = \frac{3V^2\epsilon_0\epsilon_M}{8l^3\rho_P}$ . This drags particles at a terminal velocity of  $v_{\text{DEP}} = \frac{m_P a_{\text{DEP}}}{6\pi r_P \eta}$  in a medium of viscosity  $\eta$ . Hence the time to trap a nanoparticle from the boundary of the trapping region to close to the gap between electrodes is  $t_{\text{trap}} \approx \frac{l}{v_{\text{DEP}}}$ . For a stable particle trapping, we desire  $s_{\text{trap}} = \frac{m_P a_{\text{DEP}} l}{k_B T} > 1$ , so that the potential energy of trapping field is above the thermal energy. Initially, there are  $n_P \approx C_P l^3$  particles in the trapping region, and they are trapped immediately by DEP. Thereafter, Brownian motion introduces newer particles into the trapping region at a relatively slow rate. To fill the central gap with  $Xn_P$  more particles it takes a time  $t_{\text{fill}} \approx \frac{Xl^2}{D}$ , where  $D = \frac{k_B T}{6\pi r_P \eta}$  is the diffusivity.

In Table 3.1, we present the above values for trapping nanoparticles in ethylene glycol. The trapping time for 10 nm particles is in the order of minutes, whereas the trapping of 1 nm is unstable ( $s_{\text{trap}} < 1$ ). To quicken the time to fill, another solvent with lower viscosity may be chosen. However, the other solvents listed in Table 3.2, do not stay as a drop on the chip for long because they evaporate quicker. Thus, when using a manual probe station to perform trapping between several pairs of electrodes in sequence, using solvents like toluene or ethanol can result in unwanted drop-casting before the DEP trapping can be completed.

Table 3.1. Resulting values for DEP trapping when applying  $V = 5$  V over a gap of  $l = 100$  nm in ethylene glycol of concentration 1 mg/ml at 300 K, to multiply the concentration by  $X = 10^4$  so that the trapping region is completed filled.

	$r_P = 10$ nm	$r_P = 1$ nm
$D$ (m <sup>2</sup> /s)	$1.36 \times 10^{-12}$	$1.36 \times 10^{-11}$
$a_{\text{DEP}}$ (m/s <sup>2</sup> )	$1.62 \times 10^8$	$1.62 \times 10^8$
$v_{\text{DEP}}$ (m/s)	$4.32 \times 10^{-3}$	$4.32 \times 10^{-5}$
$s_{\text{trap}}$	316	0.32 (unstable)
$t_{\text{trap}}$ (s)	$2.32 \times 10^{-5}$	$2.32 \times 10^{-3}$
$n_P$	0.01	12
$t_{\text{fill}}$ (s)	73	7

Table 3.2. Solvents for nanoparticles and their physical parameters that affect NP trapping

Solvent medium	Vapor Pressure (kPa)	Dielectric constant	Viscosity (mPa.s)
DI water	2.3	81	0.9
ethylene glycol	0.5	38	16
toluene	2.8	2.4	0.6
ethanol	5.83	24	1.2

### 3.3 Conclusion

We see from theory that using dielectrophoresis to trap 10 nm gold particles with octanethiol coating, in ethylene glycol of concentration 1 mg/ml, over a 100 nm gap between 2 electrodes, takes minutes to fill the trapping region when applying 5 V at 1 MHz. For fabricating NP clusters, we start the trapping by applying voltages on a pair of diametrically opposite electrodes for a few minutes and repeat this ritual sequentially on the other pairs.

After the DEP trapping is complete, the chip is rinsed in isopropanol to wash away the non-trapped NPs, and the residual isopropanol is blow dried by a nitrogen spray gun. We verify that trapping is successful by imaging the NP cluster using atomic-force microscopy and by measuring conductance between pairs of electrodes.

### REFERENCES

1. [Carbaugh2017] Carbaugh, D. J., Pandya, S. G., Wright, J. T., Kaya, S., & Rahman, F. (2017). Combination photo and electron beam lithography with polymethyl methacrylate (PMMA) resist. *Nanotechnology*, 28(45), 455301.
2. [Nielen2015] Nielen, R. F. (2015) *Programmable Nanoparticle Networks: working towards evolving Boolean logic in designless nanoparticle networks at room temperature* (Master's thesis). Enschede: University of Twente.
3. [Andres1996] Andres, R. P., Bielfeld, J. D., Henderson, J. I., & Janes, D. B. (1996). Self-assembly of a two-dimensional superlattice of molecularly linked metal clusters. *Science*, 273(5282), 1690.
4. [Piner1999] Piner, R. D., Zhu, J., Xu, F., Hong, S., & Mirkin, C. A. (1999). " Dip-pen" nanolithography. *science*, 283(5402), 661-663.
5. [Bernard2007] Bernard, L., Calame, M., van der Molen, S. J., Liao, J., & Schönenberger, C. (2007). Controlled formation of metallic nanowires via Au nanoparticle ac trapping. *Nanotechnology*, 18(23), 235202.
6. [Pethig2010] Pethig, R. (2010). Dielectrophoresis: Status of the theory, technology, and applications. *Biomicrofluidics*, 4(2), 022811.
7. [Li2008] Li, M., Qu, Y., Dong, Z., Wang, Y., & Li, W. J. (2008). Limitations of Au particle nanoassembly using dielectrophoretic force—a parametric experimental and theoretical study. *IEEE Transactions on Nanotechnology*, 7(4), 477-479.
8. [Castellanos2003] Castellanos, A., Ramos, A., Gonzalez, A., Green, N. G., & Morgan, H. (2003). Electrohydrodynamics and dielectrophoresis in microsystems: scaling laws. *Journal of Physics D: Applied Physics*, 36(20), 2584.

## 4 Experiments on NP clusters<sup>1</sup>

In Chapter 1, we set clear our desire to use evolution on nanomaterials to reach the ultimate physical limits of computation. Prior work in that direction was done by Tour *et al.* [Tour2002], who introduced the concept of a ‘nanocell’, which is a disordered 2-D network of NPs serving as metallic interconnects to molecules in a cell  $\sim 1 \mu\text{m}^2$  with multiple electrodes around it. Depending on the history of applied voltages, the molecules can switch between being uncharged and insulating, charged and conducting, or overcharged and insulating [Chen1999]. They proposed applying Boolean sequences as time-varying voltages on input electrodes and configure the voltages on control electrodes to realize a logic function in the measured current of an output electrode. By means of simulations, they showed that genetic algorithms could configure a nanocell into Boolean logic gates. However, experimental realization of this concept of a molecular computer has remained elusive so far.

In our work, we realize such Boolean logic gates on a cluster of 10-100 NPs separated by molecules, acting as a SET network with rich physics due to Coulomb blockade (as discussed in Chapter 2). Unlike Tour *et al.*’s proposal, the computation happens in our device due to the NPs. Figure 4.1a shows schematically the disordered network of 20 nm Au NPs interconnected by insulating molecules (1-octanethiols). The NPs are trapped in a circular region (200 nm in diameter) between radial metal (Ti/Au) electrodes on top of a highly doped Si/SiO<sub>2</sub> substrate, which functions as a back gate. At low temperatures, the NP’s charging energy  $E_C = e^2/C$  (with  $e$  the electron charge and  $C$  the NP’s total capacitance) is larger than the thermal energy, and each NP exhibits Coulomb blockade and acts as a SET (Figure 4.1b). One electron at a time can tunnel when sufficient energy is available (ON state), either by applying a voltage across the SET or by electrostatically shifting its potential. Otherwise, the transport is blocked because of the Coulomb blockade (OFF state). This disordered NP assembly therefore provides an interconnected network of robust nonlinear periodic switches as a result of the Coulomb oscillations of the individual NPs. Before we attempt to evolve Boolean logic gates as shown in Figure 4.1c, we shall study the conditions for Coulomb blockade and the extent of channels for tunnelling electrons by theoretical estimates and experimental measurements on our device.

---

<sup>1</sup> Some contents of this chapter have been previously published: Bose, S. K., Lawrence, C.P. *et al.* (2015). Evolution of a designless nanoparticle network into reconfigurable boolean logic. *Nature nanotechnology*, 10(12), 1048-1052.

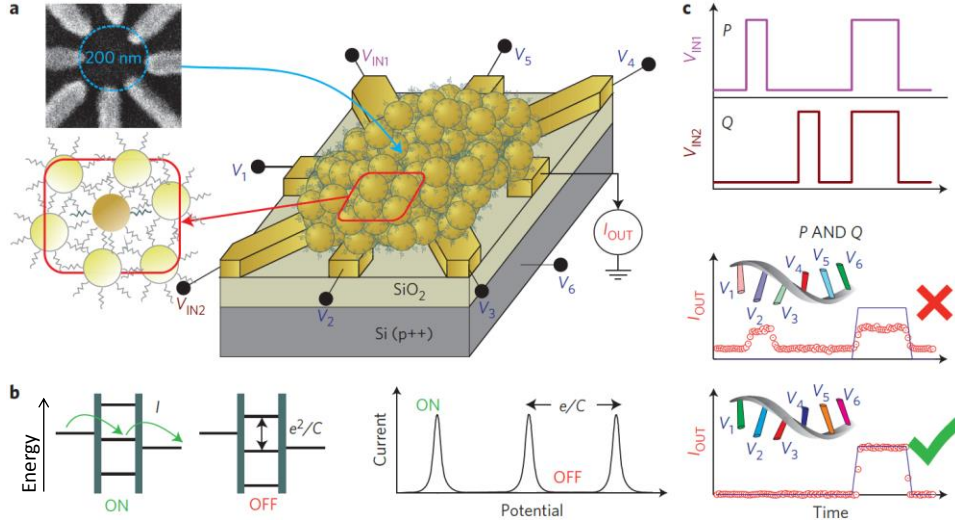


Figure 4.1. Schematic of the device layout and working principle. **a**, Disordered network of  $\sim 20$  nm Au NPs separated by  $\sim 1$  nm 1-octanethiols in between eight Ti/Au electrodes, shown in the scanning electron micrograph (top inset). Two time-varying input-voltage signals  $V_{IN1}$  and  $V_{IN2}$  and six static voltages  $V_1$ – $V_6$  give rise to an output current  $I_{OUT}$ . **b**, The NPs act as SETs at low temperatures. The potential diagrams illustrate the single-electron tunnelling (ON state) and the Coulomb blockade (OFF state). The conductance state of a single NP can be switched periodically between ON and OFF by varying its electrostatic potential. **c**, Input-voltage signals  $V_{IN1,2}$  of the form  $P, Q$  generate an output current  $I_{OUT}$  that is intended to represent  $P$  AND  $Q$ . The cases of a ‘bad’ (red cross) or a ‘good’ (green tick mark) voltage configuration  $V_1$ – $V_6$  is schematically represented by strands of genetic information shown in the insets.

#### 4.1 Coulomb blockade features

The self-capacitance  $C_{self}$  of an isolated spherical NP (radius  $r_{NP} = 10$  nm, near  $\text{SiO}_2$ ,  $\epsilon_r = 3.9$ ) is given by  $C_{self} = 4\pi\epsilon_0\epsilon_r r_{NP} = 3.9$  aF. The mutual-capacitance  $C_m$  between adjacent NPs separated by molecular barriers (1-octanethiols,  $\epsilon_r = 2.6$ ) of width  $d$  (1 nm)  $\ll r_{NP}$  is given by  $C_m = 4\pi\epsilon_0\epsilon_r r_{NP}^2 / (2r_{NP} + d) = 1.3$  aF. Looking at the atomic force micrograph of a NP cluster (Figure 4.2a) we see that the NPs are quite closely packed. Assuming 9 nearest neighbours around a NP (at the base of close-packing structure), its total capacitance  $C_0 = C_{self} + 9 C_m = 15.6$  aF. Hence, its charging energy  $E_c = e^2/C_0 = 10$  meV. For a single NP, the voltage threshold where it is in an OFF-state is given by  $eV_0 = E_c$ , i.e. charging potential  $V_0 = 10$  mV. When the thermal energy broadening  $3.5 k_B T$  is comparable to 10 meV ( $T \sim 30$  K), we completely lose nonlinear switching behaviour. In fact, we experimentally observe that Coulomb blockade is incomplete even at 5 K as shown in Figure 4.2b. We find that electron transport below  $\sim 5$  K is dominated by Coulomb blockade, and strongly depends on the input and output electrodes used, as well as on the voltages applied on the remaining electrodes and the back-gate (Figure 4.2c-e).



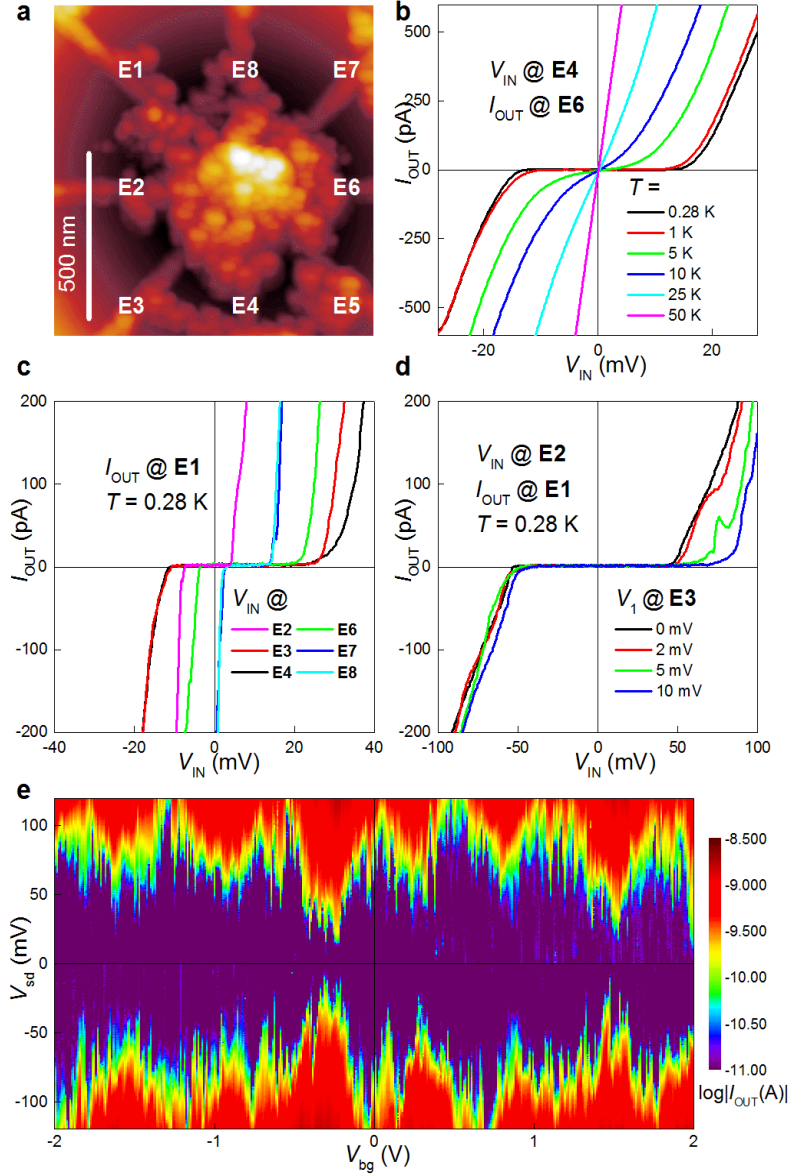


Figure 4.2. (a) Atomic force micrograph of a NP cluster with electrodes numbered E1-E8. The electron transport below  $\sim 5$  K is dominated by Coulomb blockade, and strongly depends on the used input and output electrodes, as well as on the voltages applied to the other electrodes and a back-gate. This is indicated by dependence of current-voltage relation  $I_{OUT}$ - $V_{IN}$  on: (b) operating temperature, (c) position of input electrode, (d) voltage on another electrode; and (e) Typical conductance between a pair of electrodes ( $I_{OUT}$ - $V_{sd}$ ) of the NP cluster as a function of the back-gate voltage  $V_{bg}$ .

## 4.2 Channel length and dimensionality of electron tunnelling

From the current-voltage relations between each pair of electrodes, we can estimate the channel length and dimensionality of the conducting sections of a NP cluster. A section could be a conducting 1-D chain, 2-D sheet or 3-D block of NPs. For any section between a pair of electrodes, we assume the voltage threshold  $V_{th} = nV_0$  where the channel length  $n$  is the smallest number of NPs to form a conducting path between the electrodes.

The total resistance  $R_{\text{tot}}$  of a section depends on its dimensionality, the tunnel resistance  $R_0$  and the connectivity (number of nearest neighbours). Assuming a regular connectivity, Figure 4.3a tabulates expressions of  $V_{\text{th}}$ ,  $R_{\text{tot}}$  and the power dissipated for each dimensionality. Note that the power dissipated (last column of Figure 4.3a) is an extrinsic quantity proportional to  $N$ , the total number of NPs involved. In Figure 4.3b, we scatter plot  $V_{\text{th}}$  and  $R_{\text{tot}}$  for measurements across various pairs of electrodes for several NP clusters. For the measurement with the lowest power dissipation of 4 pW at  $V_{\text{th}} \sim 10$  mV and  $R_{\text{tot}} \sim 25$  M $\Omega$ , given  $V_0 = 10$  mV we estimate  $n = 1 \Rightarrow N = 1$  and thus  $R_0 = 25$  M $\Omega$ , which is much larger than the quantum resistance  $h/e^2 = 25.8$  k $\Omega$  and thus ideal for Coulomb blockade.

In Figure 4.3b, we also estimate that in total around 10-100 NPs participate in electron tunnelling between various pairs of electrodes. The maximum channel length is  $n = 10$ , which is expected for diametrically opposite electrodes separated by 200 nm and connected by NPs of radius 10 nm. We also find sections with  $R_{\text{tot}}$  lower than even those of a 3-D block. Such ‘hyperconnected’ clusters (exhibiting a large number of nearest neighbours) may have much more than 8 participating nearest neighbours on average.

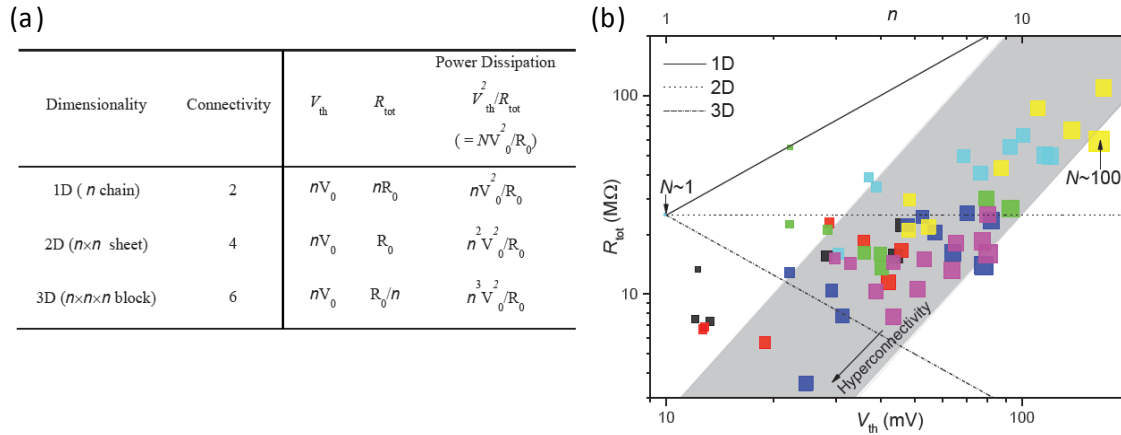


Figure 4.3. (a) Table of voltage threshold  $V_{\text{th}}$ , total resistance  $R_{\text{tot}}$ , and power dissipation for conducting sections of different dimensions and regular connectivity. The channel length  $n$  is variable, but charging potential  $V_0$  and tunnel resistance  $R_0$  are assumed to be constant. (b) Among different NP clusters (distinguished by colours), we plot a square at the experimentally measured  $(V_{\text{th}}, R_{\text{tot}})$  for various pairs of electrodes. The size of the square represents the total number of NPs involved, estimated (using last column of table) as  $N = (V_{\text{th}}^2/R_{\text{tot}})/(V_0^2/R_0)$ . The solid, dashed and dashed-dot lines indicate the dimensionality. We estimate that most electrode pairs have between 10-100 NPs participating in electron transport in between them, indicated by the grey shaded region. Clusters with very low  $R_{\text{tot}}$  are expected to be ‘hyperconnected’ (i.e. exhibiting a large number of nearest neighbors).

### 4.3 Abundance of Boolean functionality (logic and memory)

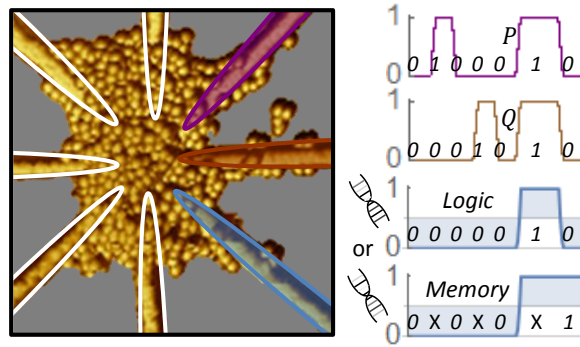


Figure 4.4. Boolean functionality in NP cluster. For an appropriate voltage configuration (control voltages on white outlined electrodes and the back-gate, input voltages on purple and brown electrodes), the output current (on blue electrode) read as a bit sequence (with values below/above a threshold current as 0/1) serves as a logic or memory function. Shown here are the logic function  $P \cdot Q$  and a memory function 0001 due to different voltage configurations, represented as strands of genetic information. The memory function is defined only by the values at  $PQ = 00$  and ignores values at X.

To realize Boolean functionality in our device (see Figure 4.4), we set the voltage configuration on a selection of control electrodes and the back-gate, and after that use two input electrodes to apply voltage signals ( $V_{IN1}(t)$ ,  $V_{IN2}(t)$ ) that map to sequences of binary variables  $P$ ,  $Q$ ) and an output electrode to measure a current signal ( $I_{OUT}(t)$ ) that is binary thresholded). The functionality (logic, memory, or noise) is determined by the input voltage amplitudes and control voltages, referred together as the voltage configuration. The pair of input sequences ( $PQ$ : 00-10-00-01-00-11-00) were chosen so that all the 4 possible cases (00, 10, 01, and 11) for Boolean logic are tested at least once. In addition, the case 00 is tested at 4 different events to record for the presence of a memory (which is any result other than ‘0000’ or ‘1111’).

Due to the limitations of our low-temperature measurement set-up, we use well-defined voltage pulse trains of total duration 1 s with rise/fall times of 30 ms and a standard hold time of 100 ms. For case 11, because both inputs are high, we take double the standard hold time so that the output current has more time to settle. The voltage signals  $V_{IN1,2}(t)$  are applied by two DACs at a sampling time  $\tau = 100 \mu\text{s}$ , and defined by the time series  $V_{IN}[a] = V_{IN}(a\tau)$ . A current amplifier converts  $I_{OUT}(t)$  to a voltage signal which an ADC measures as an output time series  $Y[a] \equiv I_{OUT}(a\tau)$  at the sampling time  $\tau$ . To test for Boolean functionality at a voltage configuration, we use an array of DACs to set voltages on control electrodes and the back-gate, allow a delay of 200 ms for the system to settle, apply  $V_{IN1,2}(t)$  and measure  $I_{OUT}(t)$ , and reset all voltages to GND. During the course of measurement, if  $I_{OUT}(t) > 1 \text{ nA}$  then voltages are immediately reset to GND, because we expect that regimes of high current are not useful for functionality based on features of Coulomb blockade. Excluding the trivial ‘always 0’ and ‘always 1’ functionalities, we have 14 different Boolean functions each for 2-input logic and 4-event memory as listed in Table 4.1.

Table 4.1. List of Boolean logic and memory functions. The symmetric logic functions are referred by their common name.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
$P$	$Q$	AND	OR	$P \cdot \bar{Q}$	XOR	$\bar{P} \cdot Q$	NAND	$\bar{Q}$	$\bar{P}$	NOR	$\bar{P} + Q$	$P + \bar{Q}$	XNOR
0101	0011	0001	0111	0100	0110	0010	1110	1100	1010	1000	1011	1101	1001

Based on  $I_{\text{OUT}}(t)$  resulting from a voltage configuration, for each functionality we may define a fitness quality  $0 \leq \bar{F} < 1$  as explained in Appendix 9. In our disordered networks, it is not clear *a priori* which voltage configurations yield top quality functions. We need to find them, for example by a random search or a genetic algorithm (GA)! We limit our search ranges of voltage configurations so that  $I_{\text{OUT}} \approx \pm 0.5$  nA at the extrema, and thus stay close to the nonlinear conductance regime. Each voltage in the search range is mapped to a gene value  $g_i \in (0,1)$ . The set of all genes,  $\mathbf{g} = \{g_1, g_2, \dots\}$ , called a genome, thus determines the voltage configuration and hence functionality.

We shall show in the next section that top-quality genomes for Boolean logic functionality could be found by means of a GA. In preparation for such experiments, it is useful to first establish by means of a random search, the abundance of functionality. For that, we consider 4 different 20 nm NP clusters (shown in Figure 4.5) measured at  $< 300$  mK, where Coulomb blockade physics is in effect to produce rich functionality. Each search lasted for  $\sim 1$  to 10 hrs and measured  $I_{\text{OUT}}(t)$  for  $\sim 3\text{k}$  to  $40\text{k}$  random genomes at a specified electrode assignment (i.e. the location of inputs, controls and output) and range of voltages. The electrode assignments for different searches are given in Figure 4.5, with electrodes 1-12 corresponding to the AFM image above, and number 0 denoting the back-gate. The expressions to obtain voltage configurations from the genes are also tabulated in the same figure.

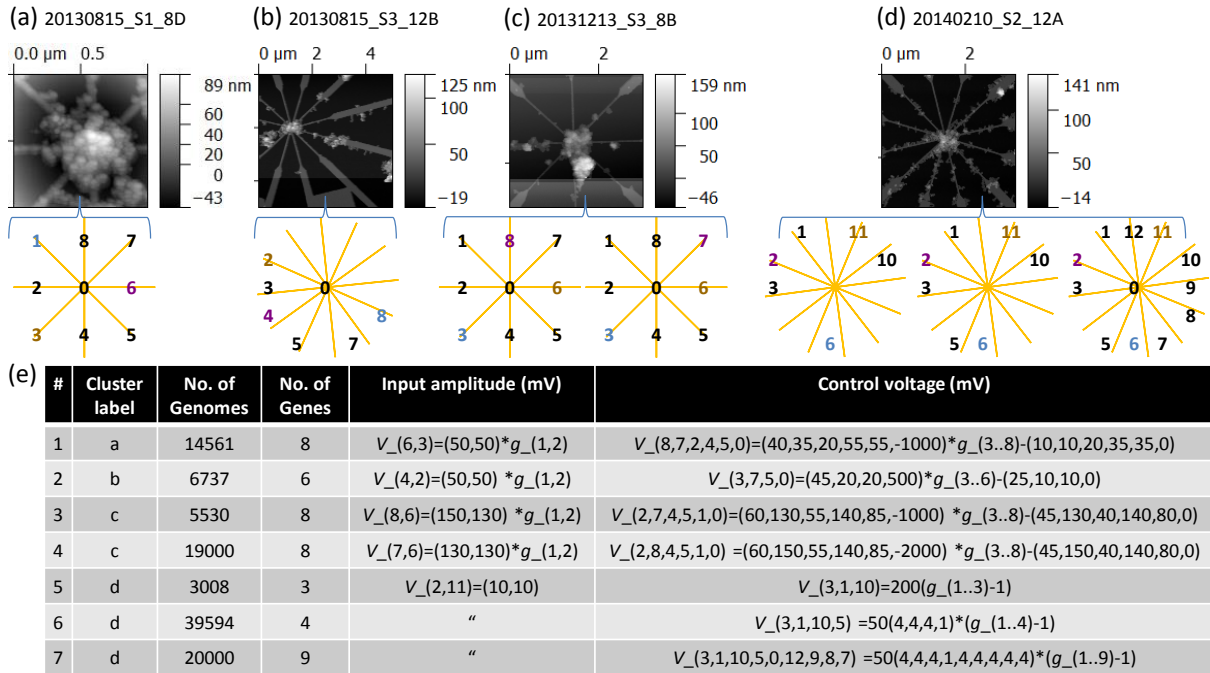


Figure 4.5. Gallery of NP clusters and their electrode assignments (as shown under a, b, c, and d) on which a set of random searches were performed with parameters tabulated in (e). Here  $x_{(i,j, \dots)}$  is understood to be the set of values  $(x_i, x_j, \dots)$ .

We assign the fitness quality of a measurement for a logic and memory function by comparing  $I_{\text{OUT}}(t)$  to an ideal output (in the time intervals of fixed inputs, excluding rising/falling edges). For each random search, we show measurements yielding the maximum fitness quality for each logic and memory function in Figure 4.6 and Figure 4.7. The threshold currents and fitness quality for these best measurements are also plotted below for reference. In order to calculate the abundance of functionality for logic or memory, each measurement in a random search is accounted for the function number (numbering according to Table 4.1) yielding maximum fitness quality (except for when all 14 fitness qualities are 0), and the accounted percentage for each function number is plotted.

The complete set of Boolean logic functions were present in all 4 clusters studied, but not for all the 7 searches due to insufficient number of genes or genomes. For searches (#1,#2,#4,#6 and #7) with greater than 6k genomes, the complete set of logic functions are present. For search #3, only the XNOR function (logic 14) was not present and the reason is likely due to its low abundance ( $< 0.1\%$  as seen for searches #2 and #4). On the other hand, for search #5, there seems to be a complete lack of logics from 5-11 and even the best measurements for logics 12-14 have rising/falling outputs during time intervals of fixed inputs. This absence of inverting logic (5-14) is likely due to insufficient number of genes. Upon introducing an extra gene to control the voltage on an electrode adjacent to the output, we find inverting logic in the same cluster, some even with greater than 1% abundance as for search #6. However, we see that search #7 on the same cluster but with 9 genes is overkill, because the average abundance for inverting logic is nearly halved.

The complete set of memory functions was also present in all 4 clusters and absent only in search #5 due to insufficient number of genes.

Next, we shall present results on the evolvability of Boolean logic gates in our NP clusters and demonstrate their robustness.

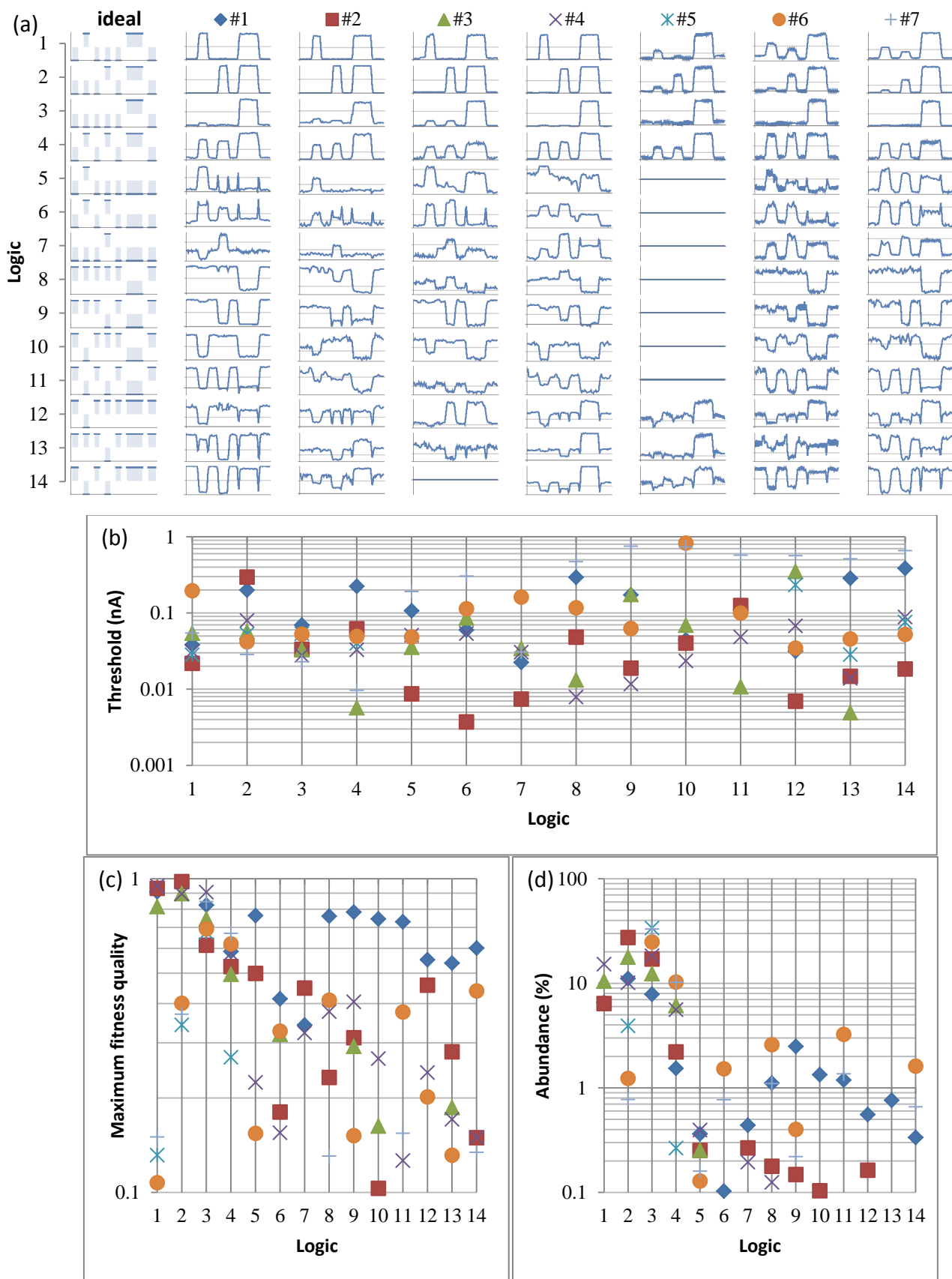


Figure 4.6. (a) Gallery of measurements showing logic functionality of maximum fitness quality for the set of 7 random searches. (b) Threshold currents and (c) fitness quality for the above measurements. (d) Abundance of logic functionality based on all measurements in a random search.

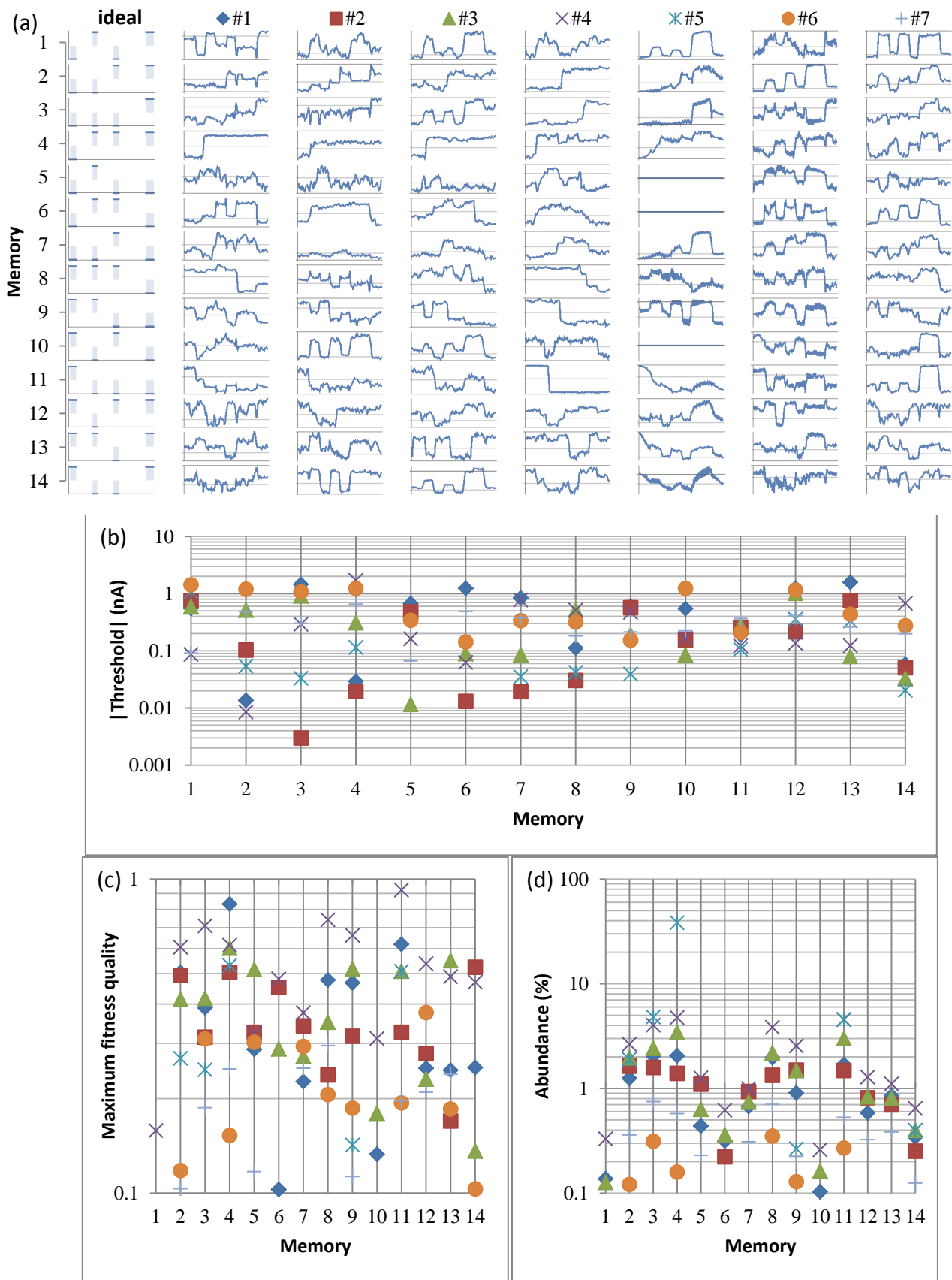


Figure 4.7. (a) Gallery of measurements showing memory functionality of maximum fitness quality for the set of random searches. (b) Absolute value of threshold currents and (c) fitness quality for the above measurements. (d) Abundance of memory functionality based on all measurements in a random search.

## 4.4 Evolution of Boolean logic gates

Apart from the blue-sky thinking that evolution in-materio can realize novel functionality by serendipity; there are also practical reasons to use evolution in our work. In our NP cluster with 8 electrodes, we first choose a pair of input electrodes and an output electrode, and seek to realize Boolean logic by optimizing the voltages on the remaining 5 control electrodes and a back-gate. Thus the voltage configuration  $(V_1, V_2, V_3, V_4, V_5, V_6 = V_{bg})$  is to be optimized in a 6-D search space, which renders a brute-force full search very time-consuming (could be order of days if a configuration is tested each second). Even with the search landscape completely unknown a priori, one could, in principle, use a conventional hill climbing (HC) search algorithm (gradient ascent/descent), which normally gives a local optimum, but can be improved by an iterated local search and simulated annealing [Kirkpatrick1983]. However, as our search landscape can have narrow hills and sometimes be noisy (explained later in Figure 4.11), we prefer to make use of artificial evolution in the form of a genetic algorithm (GA) in which laws of natural selection inspired by Darwinian evolution are utilized [Holland1992]. A GA can leap through local optima by mutations, discover improved functionality by breeding a diverse set of partial functionality [Mitchell1994], and is fault-tolerant to single points of failures if genomic diversity in the surviving population is large enough.

We draw an analogy with how genes determine the functionality of an organism, and map each configuration voltage to a gene, and hence the complete voltage configuration  $(V_1 - V_6)$  as a genome. To optimize genomes by a GA, we shall define a fitness score to evaluate a genome's desirability based on its performance (which is the measured output in our case).

### 4.4.1 Fitness score

A fitness score to guide evolution can be different from the fitness quality  $\bar{F}$  that we used in Section 4.3. This is because, a stringent requirement for quality ignores genomes that provide partial functionality (for which  $\bar{F} = 0$ ). But those genomes could serve as building-blocks for novel genomes with improved functionality, so let us define a fitness score  $F$  that rewards partial functionality.

To compute  $F$  we compare the measured output  $\mathbf{Y}$  to satisfy an ideal output  $\mathbf{X}$ , ignoring points near logic transitions as defined by weights  $\mathbf{W}$  (see Figure 4.8). We linear fit  $\mathbf{Y} = m\mathbf{X} + c$ , by estimating the magnitude  $m$  and offset  $c$  to minimize the residual error  $r$  under weights  $\mathbf{W}$ . The fitting can be done via any of the following procedures, namely least-square fit (ideal for Gaussian-distributed noise), least absolute residual fit (less sensitive to outliers) and bisquare fit (least sensitive to outliers). We choose the bisquare fit, as we assume that in a GA it is better to minimize the influence of outliers in deciding the fitness of genomes.



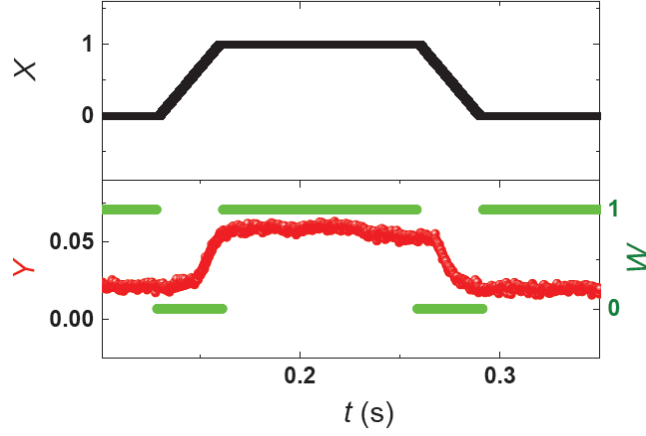


Figure 4.8. The measured output  $Y$  (red symbols in lower panel) is fit to an ideal output  $X$  (upper panel). During logic transitions, the weights  $W$  are set to 0, and the corresponding measurements are neglected.

For good output functionality, we desire a high signal-to-noise ratio ( $\text{SNR} \equiv m/\sqrt{r}$ ) and low offset to zero  $|c|$ . Combining these qualities with a mixing parameter  $\delta$ , we define the fitness score as

$$F = \frac{m}{\sqrt{r} + \delta|c|}.$$

Equation 4.1

We use an empirical value of  $\delta = 0.01$ , because it was found to work well because having a low residual error is a more complex functionality than having a low offset.

#### 4.4.2 Genetic algorithm (GA)

A GA starts with an initial generation of genomes  $\mathbf{G}_0$  and runs until a final generation of genomes  $\mathbf{G}_N$  has a satisfactory fitness distribution. Each generation  $\mathbf{G}_n$  consists of genomes  $\mathbf{g} = G_{n,i}$  where  $i = 1$  to 20, and recall that the genome  $\mathbf{g}$  is a vector of values between 0-1 that specify the voltage configuration as explained in Section 4.3. To preserve the best-performing genomes (with high fitness score), and yet search for better-performing candidates, we introduce a composite cloning–breeding approach, as depicted in Figure 4.9.

The generation  $\mathbf{G}_n$  is sorted in the order of descending fitness and the next generation  $\mathbf{G}_{(n+1)}$  is produced as follows:

$G_{(n+1),1} - G_{(n+1),5}$	=	$G_{n,1} - G_{n,5}$
$G_{(n+1),6} - G_{(n+1),10}$	=	$G_{n,i}^{+1\%} \times G_{n,i}^{-1\%} (i = 1 - 5)$
$G_{(n+1),11} - G_{(n+1),15}$	=	$G_{n,i} \times G_{n,(i+1)} (i = 1 - 5)$
$G_{(n+1),16} - G_{(n+1),20}$	=	$G_{n,i} \times \text{RandomPick}[\mathbf{G}_n] (i = 1 - 5)$

Equation 4.2

where  $G_{n,i}^{\pm 1\%}$  are obtained by modifying the genes of  $G_{n,i}$  by  $\pm 1\%$ ; and the breeding method  $G_{(n+1),i} = G_{n,j} \times G_{n,k}$  consists of a uniform crossover with a mixing ratio of 0.5 in which each gene

$g_l$  of the offspring  $\mathbf{g} = G_{(n+1),i}$  is assigned randomly from either parent gene  $G_{n,j,l}$  or  $G_{n,k,l}$ , followed by the mutation of  $g_l$  to a random value with a mutation probability  $p_M = 0.1$ .

When  $\mathbf{G}_{(n+1)}$  is complete, its genomes are tested for their fitness, the generation number is updated as  $n = n + 1$  and the above procedure is repeated until a satisfactory fitness score is reached.

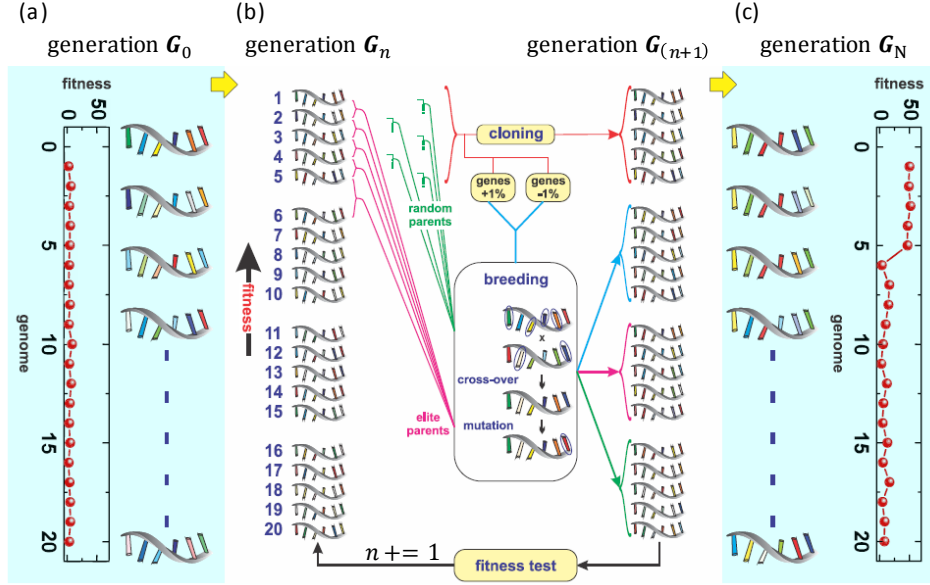


Figure 4.9. GA schematic. A genetic algorithm searches for improved genomes by iterating over many generations. Every generation  $\mathbf{G}_n$ , has a population of  $N_G = 20$  genomes (each represented by a twisted strand) that are each composed of  $N_g = 6$  genes (control voltages  $V_{1..6}$ , represented by coloured bars attached to a strand). (a) The GA starts with a random initial generation  $\mathbf{G}_0$ , whose fitness is generally low. (b) With a composite cloning-breeding procedure for evolving generation  $\mathbf{G}_n$  into  $\mathbf{G}_{(n+1)}$ , we eventually reach (c), a final generation  $\mathbf{G}_N$  whose top genomes have higher fitness than in  $\mathbf{G}_0$ .

#### 4.4.3 Universal evolvability in a compact device

Using the artificial evolution procedure described above, we succeeded to evolve all major Boolean logic gates in one and the same NP cluster at our base temperature of  $\sim 0.3$  K (as shown in Figure 4.10). These results comprise the first experimental demonstration of exploiting disordered matter at the nanoscale for computational functionality. Similar results were obtained in 3 independent NP clusters on separate chips at  $\sim 0.3$  K and on a Boron dopant cluster at 77 K in the MSc project of [vanGelder2017].

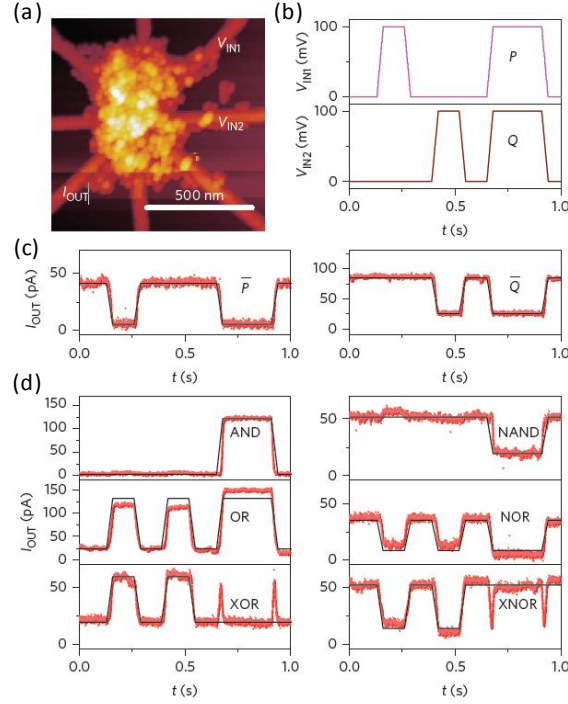


Figure 4.10. (a) Atomic force micrograph of a typical NP cluster assembled between eight electrodes. Electrodes marked  $V_{IN1,2}$  and  $I_{OUT}$  are used for the signal input and output, with the remaining electrodes and the back gate for the control voltages. (b) Time-varying input voltages  $V_{IN1,2}$  of the forms  $P$  and  $Q$ . (c) Inverter gates and (d) a complete set of basic two-input Boolean logic gates obtained by artificial evolution of the NP network at  $\sim 0.3$  K. Red symbols are experimental data and solid black lines are the matching logic.

For the exclusive gates (XOR and XNOR), spike-like features are observed at the rising and falling edges of the  $PQ = 11$  input, as expected for a finite slope in the input signals. The signal-to-noise ratio (defined as the ratio of the current that corresponds to logic 1 and the background noise floor of 5 pA) for these exclusive gates is  $\sim 10$ , whereas for all other gates it is  $\sim 20$ . Interestingly, these exclusive gates are inherently more difficult to evolve, in agreement with the theoretical framework on evolvability developed by Valiant [Valiant2009].

#### 4.5 GA search convergence speed and optima

Evolution progresses through two mechanisms: (1) breeding and (2) mutation. Breeding becomes ineffective when the population can no longer produce fitter children just by inbreeding. In such cases, mutation may cause changes that enable effective breeding again. So, although evolution never stops, per se, we say our GA converges momentarily, whenever its fitness scores stabilize. We take the fitness score of the cloned genomes,  $F(G_{n,i})$ :  $i = 1-5$ , and fit it to an asymptotic  $F(n) = F_0(1 - 0.1e^{n_0-n})$ . Here,  $F_0$  is the stable fitness score and  $n_0$  is the number of generations required to converge to 90% of stable  $F_0$ . For each logic gate, the GA almost always converged to a viable genome (defined by a fitness score much larger than one) in less than 200 generations.

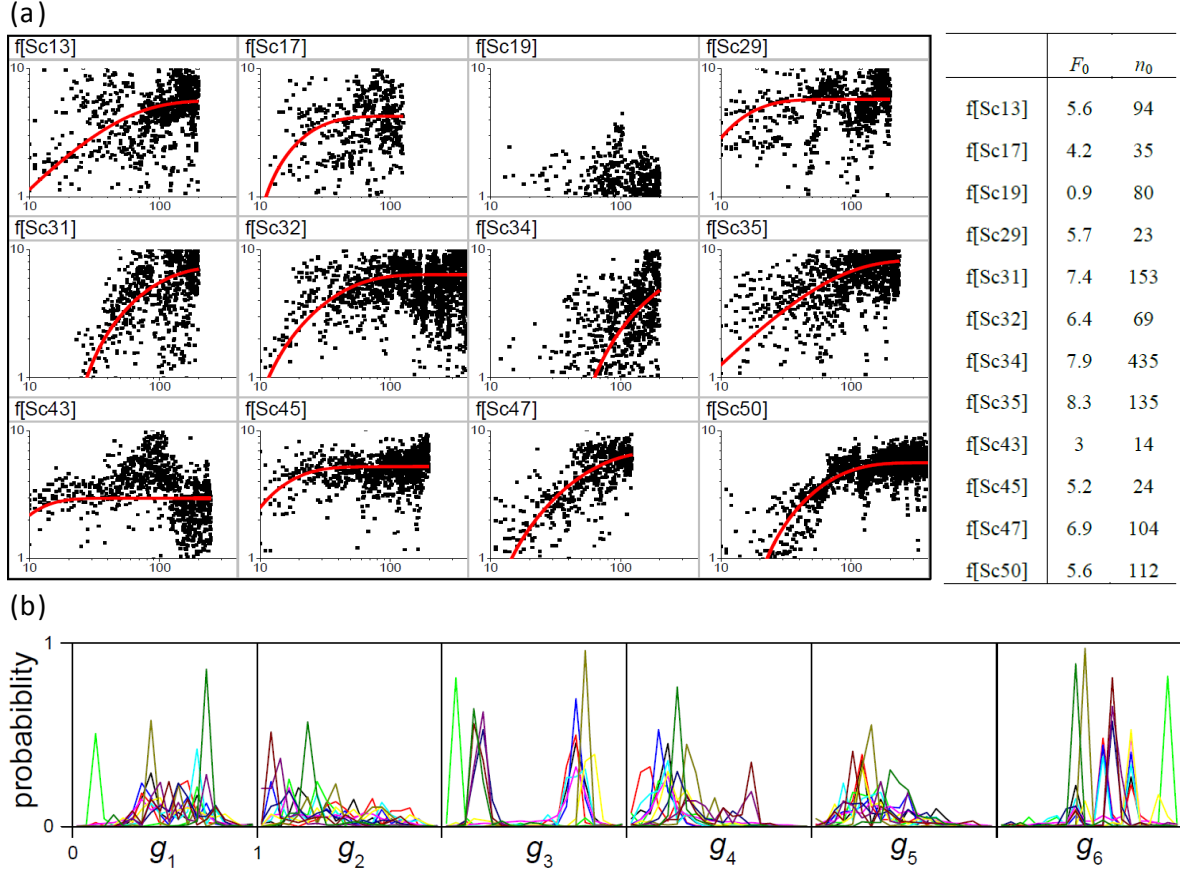


Figure 4.11. GA convergence. (a) We scatter plot  $F(G_{n,i})$ :  $i = 1-5$  vs.  $n$ , i.e. fitness scores of the cloned genomes in every generation, for multiple re-runs of GA search for a NAND. f[Sc#] corresponds to the fitness scatter of search number #, which are not contiguous as the NAND searches were interspersed with search for other gates and thermal cycles, over a period of two weeks. The red line is an asymptotic fit from which we obtain a stable score  $F_0$  and convergence generation  $n_0$ . (b) We plot a histogram (20 bins) of the experimental probability of genes explored for the above multiple re-runs (distinguished by colours).

In Figure 4.11a convergence results of the fitness scores of the cloned genomes are plotted in a log-log scale for multiple re-runs of GA search for a NAND. If we define  $F_0 > 2$  as criterion for success, we see that this GA search converged in 11 out of 12 identical cases. We see that  $n_0$  has a median value of  $\sim 90$  generations = 1800 genomes = 40 minutes of experimental measurements. By optimization of the experimental setup to work with high frequency lines at higher temperatures we can easily speed up by a factor of 100–1000 in time.

Also, the genomes have an affinity to certain gene values whose desirability is common across re-runs (see Figure 4.11b). This is especially striking in  $g_3$  and  $g_6$ , which is reasonable because  $g_3$  determines the voltage applied on an electrode adjacent to the output and  $g_6$  determines the global back-gate voltage.

If a GA with a population of  $N_G$  genomes takes  $n_0$  generations to reach a fitness  $F_0$ , a competitive search should produce at least one fitness score  $F \geq F_0$  in a sample of  $N_G n_0$  genomes. In Figure 4.12, we see that the GA is far more competitive than a random search.

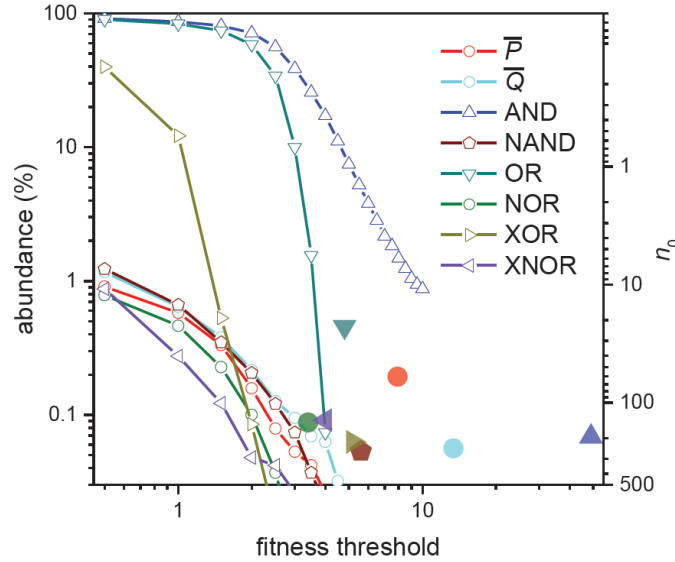


Figure 4.12. Random search vs. GA. The line plots for each logic gate correspond to the abundance of its fitness scores to be greater than a fitness threshold, in a test of 19,000 randomly generated genomes. An equivalent metric for the GA searches of each logic defined as fitness threshold =  $F_0$  and abundance % =  $100/(N_G n_0)$  is plotted as solid symbols.

## 4.6 Robustness of the device

A standard definition for robustness is "The degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions" [ISO2010]. Here we shall study the robustness of our device under varying input sequences and voltages, perturbations in voltage configuration, reconfigurations, thermal stress and ageing.

### 4.6.1 Combinatorial logic functionality

The evolved logic functionality is independent of the input waveform history, as illustrated in Figure 4.13 in which we test an input sequence that is longer than the standard 1s sequence (under which its fitness was tested during evolution).

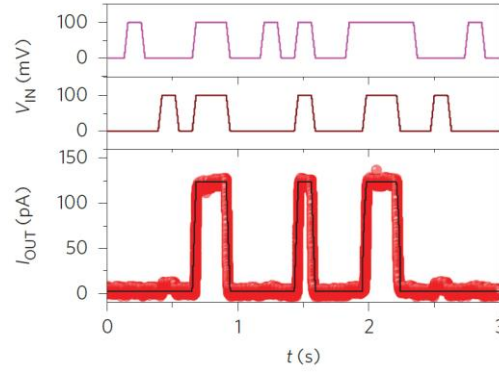


Figure 4.13. An arbitrary sequence of input signals  $P$  and  $Q$  correctly outputs  $P \cdot Q$ , which implies combinatorial functionality. The red symbols are experimental data, and the solid black line are the matching logic.

#### 4.6.2 Input voltage error tolerance

The gate functionality can extend well beyond the input-voltage levels they were evolved for. Figure 4.14 shows that a gate evolved to be an inverter for input amplitude  $V_{IN} = 100$  mV, exhibits negative differential resistance (NDR) in a range that is  $-50 < V_{IN} < 100$  mV.

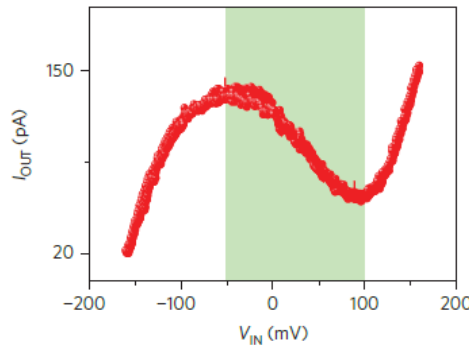


Figure 4.14. Current-voltage relation for an inverter, GA optimized for  $V_{IN} = 100$  mV, showing the NDR for the  $-50 < V_{IN} < 100$  mV range (shaded region).

#### 4.6.3 Control voltage sensitivity

The robustness of a genome's performance to voltage fluctuations may be characterized by the variation in the fitness landscape. However, exhaustively characterizing this variation in a 6-dimensional variable space is time consuming. We need an algorithm that (1) samples the space such that it captures the regions of interest, (2) predicts outcomes by weighing in observations from crucial nearest neighbours. If our goal is to estimate the robustness of the fittest genome  $G_\alpha$ , it is crucial to sample more genomes with fitness near  $F_\alpha$ . And this is what a GA search does, naturally! So, we take those genomes and predict the fitness landscape by calculating a random forests regression [Lin2006]. This technique is well suited, as the shape of the neighbourhood used by a random forest adapts to the local importance of each variable (i.e. gene).

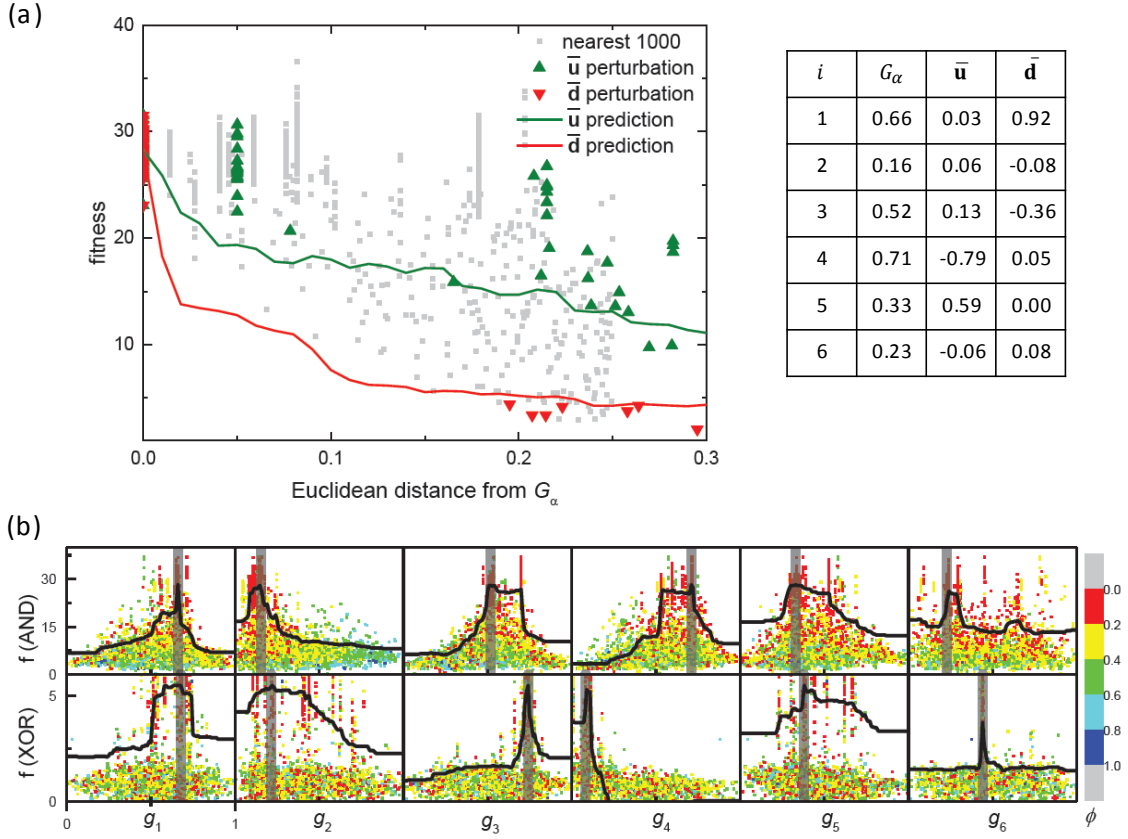


Figure 4.15. (a) Sensitivity versus perturbation direction: We plot the fitness of genomes explored in a GA search (for AND) near the fittest genome  $G_\alpha$ . The green/red triangles correspond to genomes perturbed along the direction  $\bar{u}$  /  $\bar{d}$  with angular deviation  $\phi < 0.5$ . The solid lines are predictions made by a random forest regression for fitness variation exactly along a perturbation direction. (b) Sensitivity versus logic: We plot the fitness variation of  $G_\alpha$  to perturbations along the coordinate axis  $g_i, i = 1 : 6$ . The unperturbed coordinates of  $G_\alpha$  are marked by the translucent black strips. The explored genomes are scatter plotted with a colour indicating their angular deviation from a 1-D perturbation along each coordinate axis, i.e. red genomes in plots for  $g_i$  are perturbations in  $G_\alpha$  with  $\phi < 0.2$  along the  $i$  axis. The solid black line is a prediction made by a random forest regression for fitness variation exactly along each coordinate axis.

In Figure 4.15a, we see that for perturbations local to  $G_\alpha$ , the fitness decreases with increasing distance. The decay rate of predicted fitness also depends substantially on the perturbation direction. In Figure 4.15b, we again see that the fitness is more sensitive to variations along certain directions, e.g. the fitness of  $G_\alpha$  for an AND is sensitive to a negative perturbation in gene  $g_3$ , whereas being insensitive to a positive perturbation. In case of an XOR, we see that the fitness is sharply sensitive to gene  $g_6$  in both directions. This implies that conventional hill-climbing algorithms are indeed unsuitable to find such logic gates. The back-gate voltage (gene 6) is found to have a particularly strong influence on the output signal, which is expected from its capacitive coupling to the full area of the NP cluster (as seen in Figure 4.2e).

#### 4.6.4 Stability over reconfigurations and thermal cycling

To illustrate the thermal stability and reconfigurability of the evolved logic gates solutions, the inverter ( $\bar{P}$ ) and AND gates were configured alternately, and the output was measured during a thermal cycle of warming up to 15 K and cooling down to base temperature ( $\sim 0.3$  K). For these two gates, Figure 4.16 shows as a function of temperature the variation of the fitness and the measured output (see inset). The temperature was ramped up and post a settling time of 2 minutes at each temperature step, the AND and inverter configurations were tested for fitness, multiple times. The average fitness from 30 measurements is plotted with the temperature  $T$ . We see that the fitness is stable over alternate reconfigurations at a single temperature, the fitness decreases with increasing temperature, and the maximum fitness score at base temperature is recovered upon thermal-cycling (up to 15 K). At higher temperatures, the fitness becomes negative for inverter logic because the device becomes a network of (nonlinear) resistors, and hence acts as voltage divider (which is non-inverting). Nevertheless, a weak inverting behaviour is seen even at 5 K.

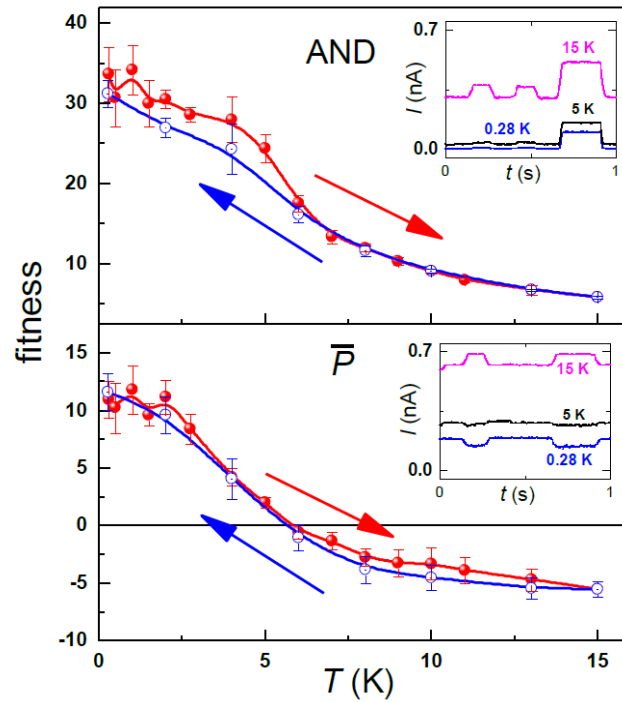


Figure 4.16. Fitness tests of inverter ( $\bar{P}$ ) and AND logic reconfigurations over a thermal cycle. The error bars represent the uncertainty in fitness for 30 measurements at each temperature. The inset shows the measured output at important temperatures. A weak inverting behaviour is seen for  $\bar{P}$  up to  $\sim 5$  K. At temperatures  $\sim 15$  K the device behaves as a nonlinear voltage divider.



#### 4.6.5 Stability over days

As long as the temperature is kept below 15 K, the logic gates are stable over several days (100 hours) as shown in Figure 4.17. However, when thermally cycling to room temperature and back, we did not find gates with (exactly) the same control voltages, so the GA search needed to be redone.

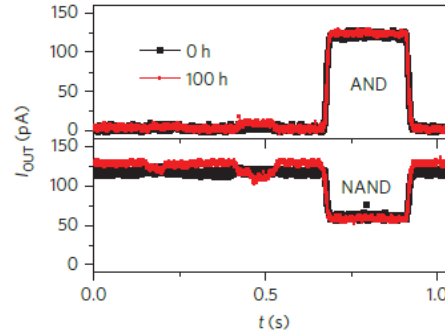


Figure 4.17. Temporal stability in measured output of AND and NAND gate configurations after 100 hours of thermal cycles under 15 K and GA searches.

#### 4.7 Multi-output binary functionality

Multi-output binary functionality can be considered as a realization of multiple Boolean logic gates in parallel. In Figure 4.18, we demonstrate a half-bit adder that was evolved by desiring the AND and XOR logic gates on two separate output electrodes. The fitness score for adder was taken to be the minimum of the fitness scores for AND and XOR.

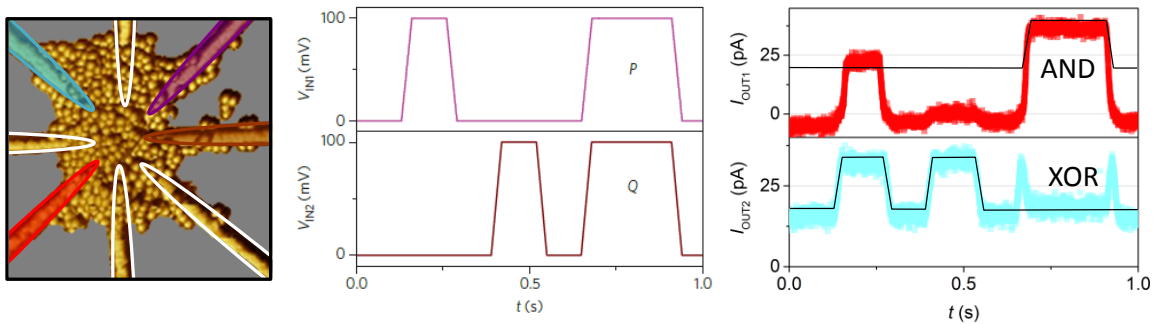


Figure 4.18. Addition functionality using a two-input-two-output gate. The input voltages  $V_{IN1,2}$  are applied on purple and brown electrodes, output currents  $I_{OUT1,2}$  are measured red and cyan electrodes. To add the input binary sequences  $P$  and  $Q$ , we realize the resulting carry and sum on the output electrodes as AND and XOR (solid black lines are the matching logic).

#### 4.8 Conclusion

We showed that a complete set of Boolean logic gates were evolvable, meaning that our physical system is computationally universal. That being said, another crucial ingredient for computation is sequential logic, for which our NP clusters may also be suitable given the abundance of Boolean

memory functions. Evolution of event-based non-volatile memory gates are interesting for synaptic computation [Abbott2004]. In addition to nonlinearity and NDR, a demonstration of (deterministic) fading memory would make our system complete for reservoir computing in materio [Dale2017]. So far, only stochastic fading-memory has been demonstrated in atomic switch networks [Demis2016,Bose2017].

If need be, there are several ways to improve our GA methods. Perhaps, the most impactful would be to explicitly reward diversity in phenomes (can be the measured output  $Y$  or its features which signify partial functionality). We may add to the fitness score a novelty factor that captures the dissimilarity of a phenome from the existing (or historical) pool of phenomes. To explore all interesting behaviours in our physical system and discover functionality by serendipity, an open-ended evolutionary algorithm such as a novelty search [Lehman2011] is also recommended.

There is no method known to be better than a GA at finding solutions in search landscapes with narrow hills and noise. However, if the landscape had broader hills, then other techniques could be viable. An interesting experiment would be to shape the landscape by controlling background charges while thermally annealing the device.

Our system meets the generally accepted four criteria to be a suitable building block (or “cell”) for the physical realization of a neural network, in particular for implementing a cellular neural network (CNN [Roska1993]) universal machine. A CNN is like a neural network, but with only nearest neighbour interactions between the cells, and therefore much easier to practically implement, but with maintaining the same power as a general neural network for solving more complex problems [Roska2005].

These four criteria are [Dogaru2003, pg. 24-26]:

- **Universality:** the possibility to use the same physical structure of the CNN cell for implementing arbitrary Boolean functions, by simply changing (programming) the cell parameter values. This is exactly what we show in our system by the reconfigurability of one and the same network into all major Boolean logic gates by evolutionarily adjusting the control voltages.
- **Compactness:** only a small set of control parameters is needed to satisfy universality. We only use six control voltages to realize all Boolean logic. The necessary area (200 nm diameter) is competitive with state-of-the-art CMOS technology for the same functionality.
- **Robustness:** the functionality must be robust against fluctuations in the control parameters. We elaborately discuss the robustness of our system against control voltage fluctuations, under thermal cycling, and as a function of time.
- **Evolvability:** the capability of a system to be evolved into a desired functionality. We clearly demonstrate this in our nanomaterials system by utilizing a genetic algorithm.

The fact that our system meets all the above criteria directly implies the applicability of our approach for realizing complex functionality beyond Boolean logic by upscaling the present architecture and using more complex electrode geometries. A technological challenge though is the

need for a large array of configuration voltages when scaling up. Thus in the next chapter, we propose an architecture where functionality is programmed by a (large) array of configuration switches that control the wiring between different NP clusters to realize functionality such a pattern recognition.

## REFERENCES

1. [Tour2002] Tour, J. M., Van Zandt, W. L., Husband, C. P., Husband, S. M., Wilson, L. S., Franzon, P. D., & Nackashi, D. P. (2002). Nanocell logic gates for molecular computing. *IEEE transactions on Nanotechnology*, 99(2), 100-109.
2. [Chen1999] Chen, J., Reed, M. A., Rawlett, A. M., & Tour, J. M. (1999). Large on-off ratios and negative differential resistance in a molecular electronic device. *science*, 286(5444), 1550-1552.
3. [Kirkpatrick1983] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598), 671-680.
4. [Holland1992] Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
5. [Mitchell1994] Mitchell, M., Holland, J. H., & Forrest, S. (1994). When will a genetic algorithm outperform hill climbing. In *Advances in neural information processing systems* (pp. 51-58).
6. [vanGelder2017] van Gelder, J. (2017). *Impurity Atom Network for Neuromorphic Computing* (Master's thesis). Enschede: University of Twente.
7. [Valiant2009] Valiant, L. G. (2009). Evolvability. *Journal of the ACM (JACM)*, 56(1), 3.
8. [ISO2010] ISO/IEC/IEEE International Standard - Systems and software engineering -- Vocabulary," in *ISO/IEC/IEEE 24765:2010(E)*, vol., no., pp.1-418, Dec. 15 2010
9. [Lin2006] Lin, Y. & Jeon, Y. (2006). Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474), 578-590.
10. [Abbott2004] Abbott, L. F., & Regehr, W. G. (2004). Synaptic computation. *Nature*, 431(7010), 796.
11. [Dale2017] Dale, M., Miller, J. F., & Stepney, S. (2017). Reservoir computing as a model for in-materio computing. In *Advances in Unconventional Computing* (pp. 533-571). Springer, Cham.
12. [Demis2016] Demis, E. C., Aguilera, R., Scharnhorst, K., Aono, M., Stieg, A. Z., & Gimzewski, J. K. (2016). Nanoarchitectonic atomic switch networks for unconventional computing. *Japanese Journal of Applied Physics*, 55(11), 1102B2.
13. [Bose2017] Bose, S. K., Mallinson, J. B., Gazoni, R. M., & Brown, S. A. (2017). Stable Self-Assembled Atomic-Switch Networks for Neuromorphic Applications. *IEEE Transactions on Electron Devices*, 64(12), 5194-5201.
14. [Lehman2011] Lehman, J., & Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2), 189-223.
15. [Roska1993] Roska, T. & Chua, L. O. (1993). The CNN Universal Machine: An Analogic Array Computer. *IEEE Trans. on Circuits and Systems-II* 40, 163-173.
16. [Roska2005] Roska, T. (2005). Cellular wave computers for brain-like spatial-temporal sensory computing. *Circuits and Systems Magazine, IEEE* 5, 5-19.
17. [Dogaru2003] Dogaru, R. (2003) *Universality and emergent computation in cellular neural networks* (Vol. 43). World Scientific.



## 5 NP internet

So far, we have shown that a disordered network of nanoparticles (in a cluster) is suitable for realizing Boolean logic functions by employing only a handful of input and control electrodes. On the other hand, for practical applications such as pattern recognition, we need to deal with megapixels of information. In this chapter, we investigate the suitability of an interconnectable network of nanoparticles (NP internet) for such advanced functionality.

### 5.1 Large-scale small-world architecture

A fundamental lesson from neuroscience [Kandel2000] is that billions of neurons connected in a small-world network [WattsStrogatz1998] are capable of intelligence. Knowledge is embedded in the wiring of synapses and decision making is via the firing of neurons. The synaptic interconnections are either excitatory or inhibitory to the firing activity of the attached neurons.

Keeping these fundamentals in mind, we propose a NP internet to realize neural network functionality using  $N_C$  nanoparticle clusters,  $N_I$  interconnect layers, a battery providing voltage bias  $V_B$  and a sensor measuring current output  $I_O$ . A compact hardware schematic for a NP internet with  $N_I = 8$  interconnect layers and  $N_C = 8$  clusters (labelled from a to h) is shown in Figure 5.1. Every cluster  $c$  has  $N_I$  interconnect electrodes, of which electrode  $i$  is connectable to the interconnect layer  $i$  by means of a switch with state  $W_{ic}$  ( $=1$  if close,  $=0$  if open). Thus, we obtain an interconnectivity matrix  $\mathbf{W}$ . The battery and sensor are also connectable to every interconnect layer  $i$  by switches  $w_{Bi}$  and  $w_{Oi}$ . Thus, we obtain a battery array  $\mathbf{w}_B$  and a sensor array  $\mathbf{w}_O$ . An example wiring configuration is shown in Figure 5.2a. Depending on the state of  $(N_C + 2)N_I$  switches, the NP internet can be one of the  $2^{N_C N_I + 2N_I}$  possible wirings.

The functionality is scalable by simply fabricating more number of NP clusters in parallel. If each NP cluster occupies an area of diameter 300 nm, then a chip of diameter 1 cm can fit about  $\left(\frac{10^{-2}}{3 \times 10^{-7}}\right)^2 \approx 10^9$  clusters. The wiring configurations are proposed to be evolved by controlling the switches using a genetic algorithm running on a computer server. A PCB has been developed for this purpose by Martin Siekman and Kesari Krishna [Krishna2017], employing CMOS-based ICs with switching times of 200 ns and leakage currents of few pA. The communication from the server to the switches can be either wireless or follow message passing through a hierarchical topology of routers designed to work at the nanoscale [Bush2015]. By employing routers with a branching factor of 64, a server can control using 5 levels of hierarchy up to  $64^5 \approx 10^9$  switches. Some redundancy may be introduced so that communication is fault-tolerant and for efficiency only the relative change in the state of switches (mutations) may be passed from the server to the NP internet.

The maximum number of tunnelling events for any path from the battery to the sensor is proportional to the depth of our network, which is fixed at  $N_I = 8$  layers, thus satisfying the small-world property. As shown before, NP clusters are capable of nonlinear and negative differential resistance behaviour, thus also satisfying the requirement of excitation and inhibition.

If the currents through a NP cluster can be modelled as a deterministic function of the voltage on its electrodes, then each wiring configuration of the NP internet realizes a recurrent neural network (RNN) whose internal states are the electrode voltages as shown in Figure 5.2b. RNNs have the capacity to deal with time-varying signals and implementing such functionality by our NP internet is discussed towards the end of this thesis. In this chapter, we focus on obtaining a stable current output to perform pattern recognition, and thus it is illustrative to reduce the RNN into a simpler feed-forward neural network (FNN) as shown in Figure 5.2c.

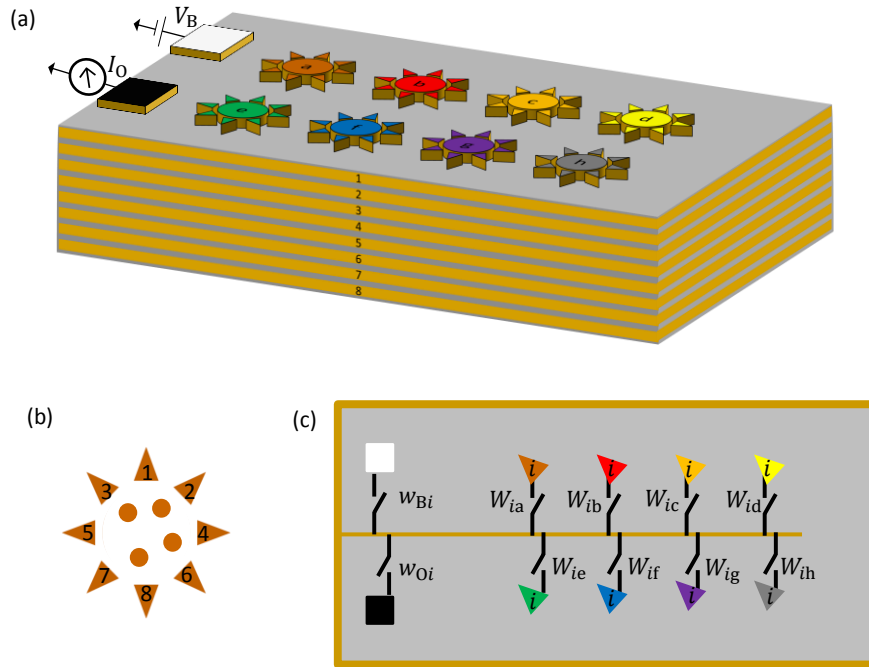


Figure 5.1. (a) A NP internet with a battery (at white electrode), current sensor (at black electrode) and several nanoparticle clusters (coloured with triangular electrodes) shown on top of interconnect layers (numbered from 1 to 8). The electrodes on top are connectable to every layer by means of switches (hidden in between layers). As a convention, first-layer is always connected to battery and last-layer is always connected to sensor. (b) Electrode numbering in a cluster, so that the first interconnect is opposite to the last (interconnect 8). (c) The wiring scheme uses switches to multiplex an interconnect layer ( $i=3$ ) to the battery, sensor, and cluster electrodes of the same number.

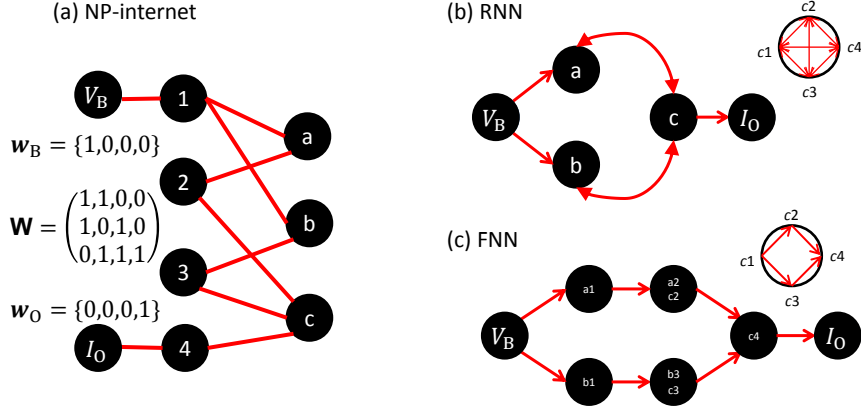


Figure 5.2. Representation of networks. (a) A NP internet defined by its wiring configuration. (b) RNN showing interactions. (c) FNN constructed by assuming a directional flow of current within the clusters  $c$ .

## 5.2 SET network model

In order to faithfully simulate real nanoparticle networks, capacitance and resistance matrices need to be computed by finite-element analysis [Banyai2015]. Here, for simplicity, we model tunnel junctions with a capacitance number  $c$  and a conductivity number  $\sigma$  such that the capacitance  $C = 0.16c$  aF, and resistance  $R = \frac{h}{\sigma e^2} = 25 \text{ k}\Omega/\sigma$ . As a reference,  $c = 1$  yields a charging energy of 1 eV which is considerably higher than the thermal noise of 25 meV at room temperature. We desire  $\sigma \ll 1$  for the electrons to be localised within the particle islands. For a nanoparticle of radius  $r_p$ , we assume that the gate junctions are purely capacitive with  $c_{GP} = r_p$ ,  $\sigma_{GP} = 0$ . The island-island junctions are defined by  $c_{\alpha\beta} = \frac{r_p^2}{d_{\alpha\beta}}$  and  $\sigma_{\alpha\beta} = \exp(-\frac{d_{\alpha\beta}}{r_T})$ ; where the distance  $d_{\alpha\beta} = r_p + |\mathbf{r}_\alpha - \mathbf{r}_\beta|$  with  $\mathbf{r}_\alpha$  the position of island  $\alpha$ , and  $r_T$  is the tunnelling length which can be estimated by solving Schrodinger's equation as  $r_T = \hbar/\sqrt{2m_e\Delta E}$ .  $\Delta E$  is the energy barrier for electron tunnelling and is dependent on the work function, charge occupancy and size of the nanoparticle [Kalered2017]. For  $\Delta E = 40$  mV we have  $r_T \approx 1$  nm.

A NP internet of  $N_C = 8$  clusters, where each cluster is a disordered network with  $N_p = 4$  nanoparticles (of radius  $r_p = 1$ ) surrounded by  $N_I = 8$  interconnect electrodes is shown in Figure 5.3. The particle positions  $\mathbf{r}_p$  are randomly distributed in a circle of radius  $N_T r_T$  and the interconnect positions  $\mathbf{r}_i$  are spaced evenly along the circumference. For every interconnect layer  $i$ , we assign an island whose capacitance number is  $c_{Gi}$ . There are  $N = N_p N_C + N_I$  islands in the entire SET network.

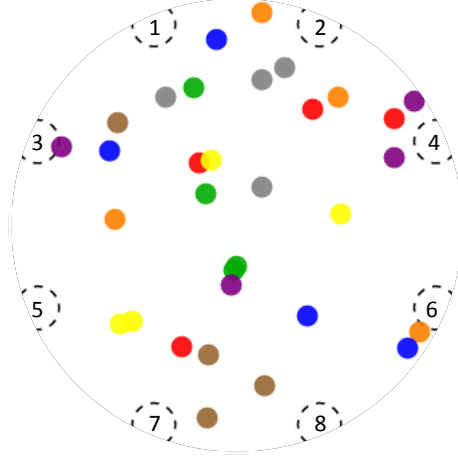


Figure 5.3. Combined representation of particles in a disordered NP internet. Each colour corresponds to a cluster of 4 NPs that is encircled by 8 interconnect electrodes.

In practice, the interconnect capacitances could be  $c_{Gi} \gg 1$  and hold 1000's of excess electrons. But charging large capacitors by single-electron tunnelling costs a lot of simulation time, so we assume that  $c_{Gi} = 10$ , a number that is just high enough for the interconnect islands to not be in Coulomb blockade. The capacitances and tunnel resistances are to be obtained from the positions  $\mathbf{r}(\alpha)$  of the islands and the wiring configuration. The bias junctions on interconnect islands are defined as  $c_{Bi} = 10w_{Bi}$  and  $\sigma_{Bi} = 100w_{Bi}$ . Similarly,  $c_{Oi} = 10w_{Oi}$  and  $\sigma_{Oi} = 100w_{Oi}$ . The interconnect-particle junction is defined as  $c_{ip} = \frac{r_p^2}{d_{ip}} W_{ic(p)}$  and  $\sigma_{ip} = \exp(-\frac{d_{ip}}{r_T}) W_{ic(p)}$ . For most of our analysis, the default parameters used are summarized in Table 5.1. In some cases, spatially varying background charge distributions are considered by adding random values between 0 and  $\mu_{BG}$  to the background electron number of each island.

Table 5.1. Default parameters for constructing a NP internet.

$N_P$	$N_C$	$N_I$	$r_P$	$r_T$	$N_T$	$\mu_{BG}$
4	8	8	1 nm	1 nm	10	0

### 5.3 Simulations of NP internets at interesting wiring configurations

Even for small numbers of  $N_C = N_I = 8$ , we have  $2^{N_I N_C + 2N_I} \approx 10^{24}$  wiring configurations in total. If we only connect the battery and sensor to the first and last interconnection respectively, then there are  $2^{N_I N_C} \approx 10^{19}$  configurations in total. Each of these configurations can be addressed as a group of  $N_C$  numbers taken from 0 to  $2^{N_I} - 1$  (e.g. 192.168.0.1.14.139.160.194).



However, configurations can be degenerate in terms of current-voltage relations. For example, if a wiring configuration is seen as an undirected graph, then all edges that cannot be part of a path from the battery to the sensor may be pruned without affecting overall current flow. More degeneracy arises when the nanoparticle clusters can be modelled by symmetric functions. For example, when clusters realize identical functions, reordering the numbers in a wiring address does not change any functionality. So by only counting wiring addresses with numbers in ascending order, we have  $\sum_{k=1}^{N_C} \binom{2^{N_I}}{k} \binom{k}{N_C-k} \approx 10^{14}$  configurations. Furthermore, if the cluster function is invariant upon mirroring or rotating of electrodes, then even more degeneracy occurs. In any case, note that configurations with the same current-voltage relations can still be useful for different current-subconfiguration functionality. So degeneracy does not necessarily mean a loss of functionality for pattern recognition, but greatly reduces the total number of fundamental configurations that need to be simulated in order to predict functionality. It is not clear *a priori* to what extent this degeneracy applies to disordered NP internets. Degeneracy in terms of current-voltage relations may be qualitative instead of the quantitative degeneracy possible in well-ordered NP networks.

To begin with a study, let us construct a subset of fundamental configurations which are *interesting*. Consider a regular graph of degree  $N_C = 8$  and vertex count  $N_I = 8$ , where each vertex  $i$  represents an interconnect layer  $i$  and each coloured edge  $e(i, j) = c$  represents a NP cluster  $c$  between interconnect electrodes  $i, j$  as shown in Figure 5.4a. Each path from 1 to 8 in this completely-coloured graph represents a wiring configuration as illustrated in Figure 5.4b. Here, every path of length  $k$  is a permutation of  $k-1$  numbers between 1 and 8. Thus there are  $\sum_k \frac{6!}{(7-k)!} = 1 + 6 + 30 + 120 + 360 + 720 + 720 = 1957$  interesting wiring configurations.

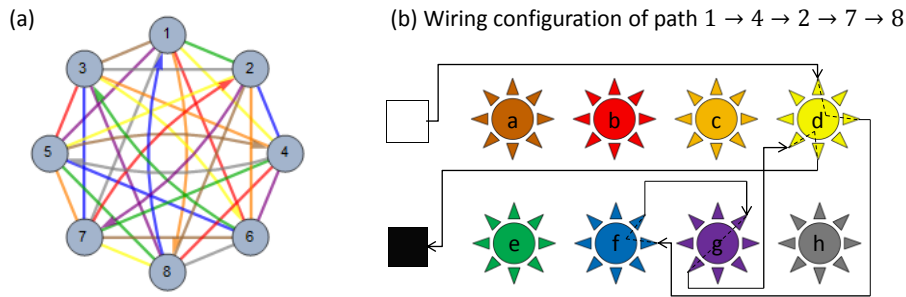


Figure 5.4.(a) A graph where every vertex has edges for the complete set of colours. Some edges are directed so that every path in the graph produces a unique wiring configuration. (b) An example wiring configuration derived from a path.

### 5.3.1 Diversity in current-voltage relation

A large diversity in relations is a necessary (but not sufficient) condition to classify a large set of data into different patterns. Hence, to study factors that improve diversity in the current-voltage relations, we construct disordered NP internets as described before and compare simulations across the set of 1957 interesting wiring configurations.

We simulate the output current  $I(V)$  for  $V = 0$  to  $1$  V in steps of  $10$  mV, by a Monte Carlo method that runs until either  $1\%$  precision or a maximum of  $1000$  tunnel events. We can confirm that our simulation results are qualitatively reproducible, by comparing across independent simulation runs on a disordered network without background charge as shown in Figure 5.5a. Moreover, we can visually see that the functionality is diverse across the wiring configurations. This diversity becomes greater for a disordered network with spatially varying background charges as shown in Figure 5.5b. Of course, the functionality is also diverse across different disordered networks as shown in Figure 5.5c and Figure 5.5d.

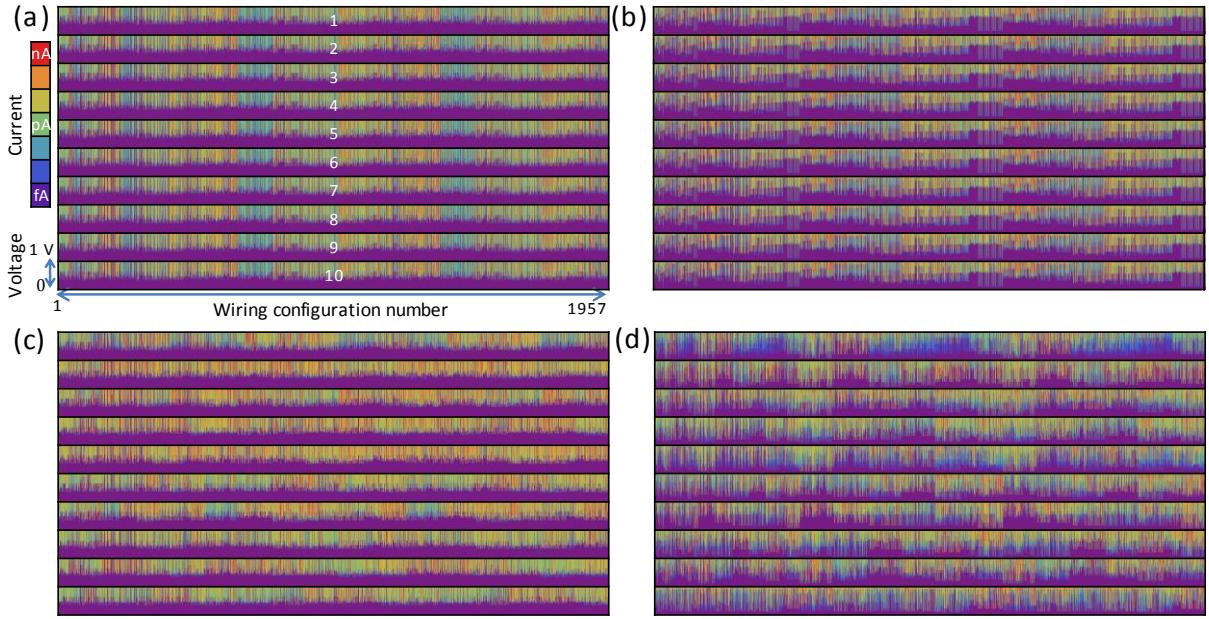


Figure 5.5. Current-voltage relations across the interesting wiring configurations shown for, (a) 10 independent simulation runs of a disordered network without background charge, (b) 10 independent simulation runs of a disordered network with  $\mu_{BG} = 1$ , (c) simulation runs of 10 different disordered networks without background charge, and (d) simulation runs of 10 different disordered networks with  $\mu_{BG} = 1$ .

In order to easily understand the functionality in these networks, we reduce the simulation result  $I(V)$  into two basic measures of functionality: the maximum current and the threshold voltage (which is defined as the voltage span between  $0$  and  $1$  V where the current is less than a noise floor of  $10$  pA).

In Figure 5.6, for each of the  $1957$  interesting wiring configurations, measures of functionality from  $10$  independent simulations of a disordered network (as in Figure 5.5a) is overlay plotted against their average. We see that: 1) there is a strong correlation between measures of independent simulations to the average among simulations, which means stochastic or multi-stable processes are a minority, and thus we can assign a unique functionality for a majority of the configurations, 2) the measures have a broad range, which highlights how diverse the functionality is across the interesting wiring configurations.

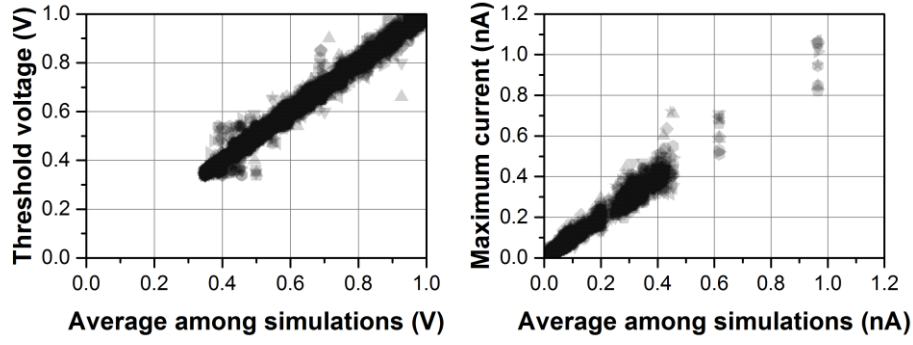


Figure 5.6. Measurement statistics of interesting wiring configurations for 10 independent simulations of a disordered network.

In Figure 5.7a-b, for each of the 1957 wiring configurations, measures from 10 different networks is overlay plotted against their average. We see that: 1) there is poor correlation between the measures of individual networks to their average among networks, which means disorder breaks degeneracy (functionality at a wiring configuration is sensitive to disorder in the system) and thus we can expect functionality to be diverse across all wiring configurations, 2) the diversity is greater in case of spatially random background charges.

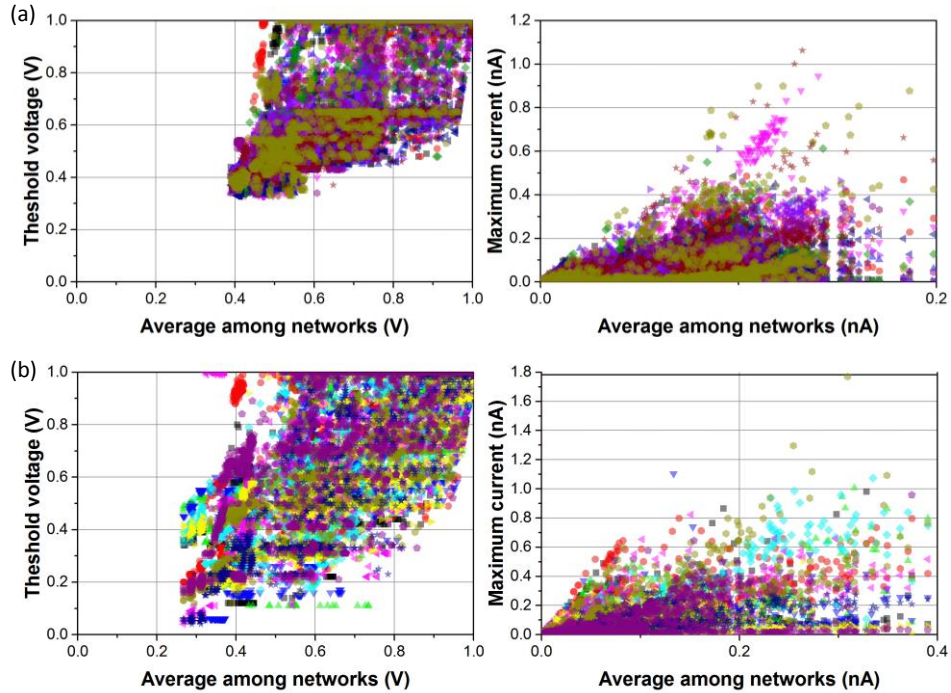


Figure 5.7. Measurement statistics of interesting wiring configurations for different disordered networks, (a) without background charge and (b) with  $\mu_{BG} = 1$ .

### 5.3.2 Inhibitory mechanism

We define inhibition as the decrease in output current after making an interconnection, or equivalently an increase in output current after breaking an interconnection. We qualify it as complete inhibition if

there is a transition from a steady current flow to a Coulomb blockaded SET network upon making an interconnection, otherwise it is partial inhibition. Furthermore, inhibition in the output current from a cluster can be either directly due to an interconnection in that cluster or indirectly due to interconnection in a different cluster that is coupled in series.

To study these inhibitory mechanisms in action, we shall search for wiring configurations of the NP internet that provide an effective hidden layer (see Figure 5.8a) between the voltage bias and output to realize XOR functionality. I.e., the current output for an input layer connectivity of  $PQ = 11$  is required to be lower than the current outputs for inputs 10 and 01. We shall restrict our search again to those NP internets with the input, output, and hidden layer connectivity derived from interesting wiring configurations as follows. The input connections  $P$  and  $Q$  are given to the beginning and ending clusters in the path of an interesting wiring configuration. The output connection is the ending connection of the path, thus to the same cluster as  $Q$ . The hidden layer connectivity is given by the interesting wiring configuration minus the beginning and ending connections of its path. Thus, interesting wiring configurations which begin and end in different clusters, a total of 1566 configurations, can be tested for XOR functionality ( $!PQ+!QP$ ) and hence study both direct inhibition ( $!Q$  when  $P=1$ ) and indirect inhibition ( $!P$  when  $Q=1$ ).

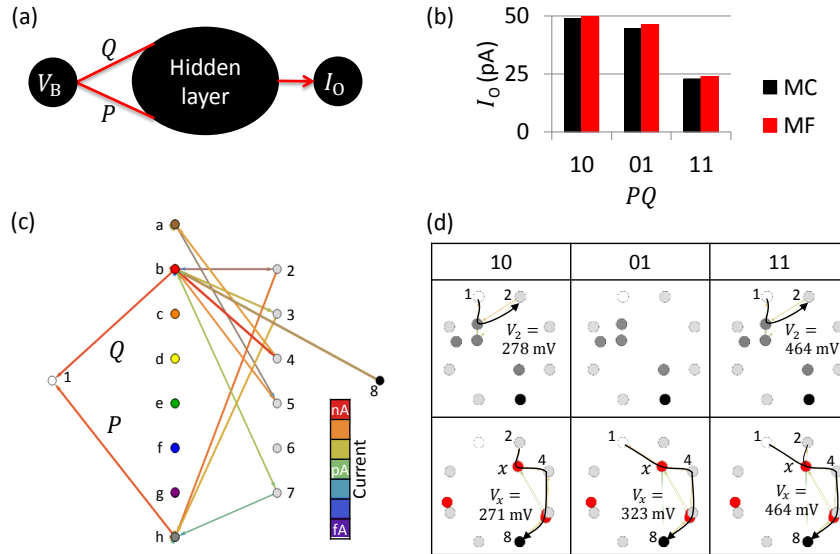


Figure 5.8. (a) An abstraction of the NP internet whose output current is a function of the input connectivity  $P$  and  $Q$ . (b) The best XOR functionality found in our search. (c) The connectivity graph of our best XOR network. The colour nodes are different clusters, the numbered nodes are interconnects and the edges are coloured to represent the current flow in the equivalent resistor network. (d) Major current flow paths and mean voltages in the input clusters vs. the input connectivity. Numbers are next to interconnecting electrodes and  $x$  denotes the most active island in the red cluster b.

We tested 100 disordered networks (without background charge) for functionality at  $V_B = 1$  V by MC simulations to a maximum of  $10^5$  tunnel events. The best XOR was found to have up to 50% inhibition, and it was cross-validated by a MF simulation taking  $\sim 200$  steps on average to reach under

1% precision. In Figure 5.8b, we present those simulation results. In Figure 5.8c, we show the NP internet with both inputs connected and current flow expected in the equivalent resistor network obtained by ignoring charging and tunnelling physics in the islands. However, the actual flow of current is much lower due to Coulomb blockade effects, and the major current flow paths and mean voltages are shown in Figure 5.8d. Input connectivity of case 11 has conducting paths of both the cases 10 and 01. However, the current flowing through the most active island  $x$  is much lower in case 11, because the mean voltage differences across the electrode-island junctions  $2|x$  and  $1|x$  are greater in cases 10 and 01 respectively.

## 5.4 Pattern recognition

In this section, we shall use simulations to study the feasibility of evolving the wiring configuration of NP internets to work as a neural network and thus perform pattern recognition [Fogel1990]. In particular, we attempt to build a classifier [Kotsiantis2007] to distinguish between binary images of digits **1-8** as shown in Figure 5.9a.

For this purpose, we split the interconnectivity matrix  $\mathbf{W}$  into input, output, and hidden layers as shown Figure 5.9b. The first 2 rows of  $\mathbf{W}$  are mapped to the input image (as binary values  $a_1, b_1, \dots, g_2$ ) such that  $W_{1a} = a_1 = 1$  if the pixel corresponding to  $a_1$  is black, and so on. We set  $\mathbf{w}_B = \{1, 1, 0, 0, 0, 0, 0, 0\}$  so that the battery is only connected to the input layer and set  $\mathbf{w}_O = \{0, 0, 0, 0, 0, 0, 0, 1\}$  so that the last row of  $\mathbf{W}$  forms the output layer. The remaining elements of  $\mathbf{W}$  form the weights of the hidden layer. For example, the output node  $h_8$  can recognize the digit **1** if there is a flow of current after setting  $W_{8h} = 1$  only if the input image looks like a 1. In a similar fashion, if the nodes  $g_8$  to  $a_8$  can recognize the digits **2** to **8**, then that NP internet is a successful classifier of these digits.

An example hidden layer functionality to perform classification based on Boolean logic is given in Figure 5.9c, and its realization on a NP internet can be as shown in Figure 5.9d. We intend to optimize the hidden layer connectivity of our NP internets using genetic algorithms to realize similar functionality. Before embarking on such an evolution, we first study the abundance of functionality in our NP internets.

### 5.4.1 Abundance of functionality

In our NP internet, an implementation of 8-digit classification could be that the flow of current is higher than a threshold only when the input-output pairs are matched as **1** –  $h_8$ , **2** –  $g_8$ ... **8** –  $a_8$ . For classification of this kind, if the current values for all the input-out pairs are visualized as a matrix, all the diagonal values must be greater than the non-diagonal values. Hence the diagonal values must all be greater than a ‘global threshold’. This condition seems too hard, even for systems with great diversity, as the abundance of functionality is low when searching across random  $8 \times 8$  matrices (see blue bars in Figure 5.10). Hence, we shall look at an easier condition that doesn’t enforce a global threshold. I.e. Each output node is said to recognize the input digit for which the output current is maximum. In this way, the abundance of functionality is much higher (see red bars in Figure 5.10). As more than 0.1% of random  $8 \times 8$  matrices can implement the easier kind of classification.

We look at NP internets with random hidden layer connectivity, and simulate the current flow for all  $8 \times 8$  pairs of input and output connectivity as given in Figure 5.1a-b. Currents are estimated at  $V_B = 0$  to 1 V, in steps of 10 mV by MC simulations of the SET networks for randomly generated NP internets with random hidden layer connectivities (relevant code given in Appendix 1). We find that



the abundance of functionality for these SET networks is comparatively low (see green bars in Figure 5.10). However, SET networks still have much better abundance than the equivalent resistor networks of NP internets (see purple bars in Figure 5.10).

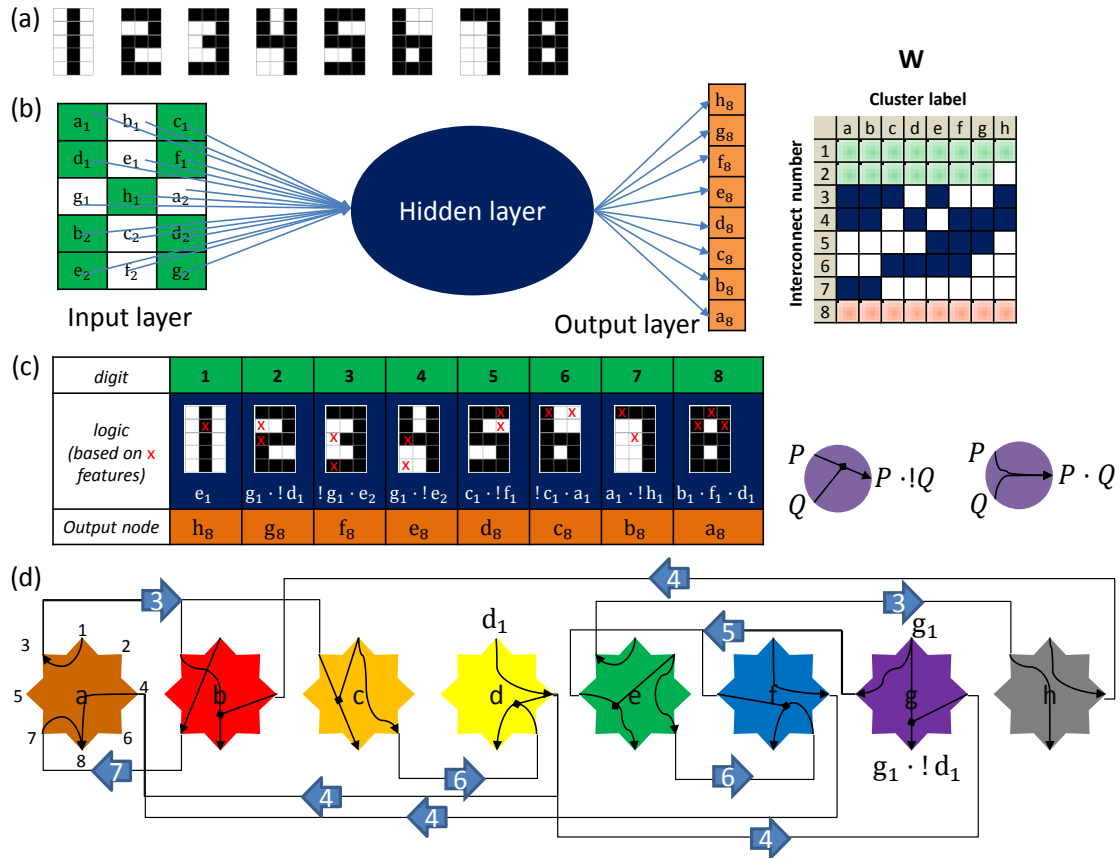


Figure 5.9. (a) Binary images of digits, represented as 3×5 pixels. (b) Mapping interconnectivity matrix of NP internet to the input image and weights of a neural network. (c) An example table of Boolean logics to recognize digits uniquely at output nodes. Just 2 kinds of logic functions are sufficient for this, as depicted aside. (d) Realization of the above functionality in our NP internet. Each colour star represents a cluster with 8 electrodes (as numbered in cluster-a) and the thick blue arrows represent flow of binary values through interconnects, always between electrodes with the same number, as indicated by the white number inside the blue arrow.

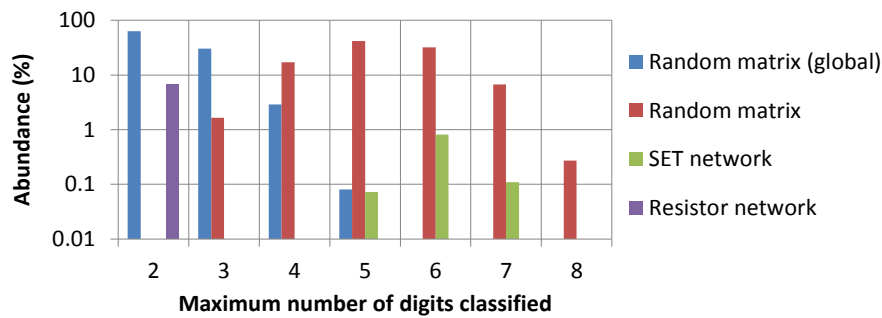


Figure 5.10. Abundance of functionality enforcing a global threshold is shown for (in blue)  $10^4$  different random 8×8 matrices. Abundance of easier functionality is shown for (in red)  $10^4$  different random 8×8 matrices, (in green) 100 different SET networks over 100 voltages and (in purple)  $10^4$  different resistor networks.

## 5.5 Conclusion

The concrete results of this chapter are the conception of the NP internet for pattern recognition and a demonstration of inhibition via simulations of interesting wiring configurations to realize XOR functionality. An experimental realization of this is recommended before embarking on more advanced functionality such as pattern recognition.

Although we showed that the current-voltage relations of our NP internets are diverse over interesting wiring configurations, that diversity is still lower than random distributions. Thus, the abundance in functionality for pattern recognition is also much lower. A sure way to increase the abundance is to increase the ‘intelligence’ of our NP internet by increasing the number of clusters  $N_C$ . Another way is to increase the intelligence of our nanoclusters by establishing optimal parameters for  $r_T, N_T, \mu_{BG}$ . With that in mind, we shall begin our next chapter on measuring the IQ of a physical system.

It may turn out that a nanomaterial system has its maximal IQ only at low temperatures, so for that purpose Suzanne Vernimmen and Max Krakters have designed a helium-tight dipstick chamber to house the NP internet [KraktersVernimmen2016]. Advantages of a vacuumable chamber are the prevention of moisture and rapid cooling that could be detrimental to solid-state electronics, and a dielectric constant = 1 so that the charging energy is maximal for a given size of NP.

## REFERENCES

1. [Kandel2000] Kandel, E., Schwartz, J., & Jessell, T. (2000). Principles of Neural Science.
2. [WattsStrogatz1998] Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *nature*, 393(6684), 440-442.
3. [Krishna2017] Krishna, K. (2017). *The Nanoparticle Internet: Towards Scalable Evolvable Hardware* (Bachelor internship report). Enschede: University of Twente.
4. [Bush2015] Bush, S. F., Paluh, J. L., Piro, G., Rao, V., Prasad, R. V., & Eckford, A. (2015). Defining communication at the bottom. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 1(1), 90-96.
5. [Banyai2015] Banyai, D. R. (2015). *Multiscale examination and modeling of electron transport in nanoscale materials and devices* (PhD thesis). Michigan Technological University.
6. [Kalered2017] Kalered, E., Brenning, N., Pilch, I., Caillault, L., Minéa, T., & Ojamäe, L. (2017). On the work function and the charging of small ( $r \leq 5$  nm) nanoparticles in plasmas. *Physics of Plasmas*, 24(1), 013702.
7. [Fogel1990] Fogel, D. B., Fogel, L. J., & Porto, V. W. (1990). Evolving neural networks. *Biological cybernetics*, 63(6), 487-493.
8. [Kotsiantis2007] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160, 3-24.
9. [KraktersVernimmen2016] Krakters, M., Vernimmen, S. E. I. (2016). *Working towards the Evolutionary programming of a NanoParticle internet at liquid nitrogen temperature* (Bachelor’s thesis). Enschede: University of Twente.



## 6 IQ of a physical system

In our quest to realize intelligence by evolving nanomaterial networks, it is useful to have a quick estimate about the abundance and evolvability of functionality for tasks like Boolean logic (Chapter 4 Section 3) and pattern recognition (Chapter 5 Section 4). To that end, here we introduce a measure for the *Intelligence Quotient* (IQ) of physical system based on its input-output relations across configurations.

### 6.1 Defining intelligence

There are many approximate ways to define intelligence [LeggHutter2007]. Albert Einstein once said that the true sign of intelligence is not knowledge, but imagination. Stephen Hawking said that intelligence is the ability to adapt to change. And now, my dad says that intelligence is the ability to understand abstract things. These opinions highlight that an intelligent system has a capacity to create new information, evolve and recognize patterns.

A few notable measures of intelligence can be based on: 1) the degree of consciousness ( $\Phi$ , as per the *integrated information theory* [Oizumi2014,Tegmark2015]); 2) the ability of an agent to succeed in a wide range of environments (*universal intelligence*, as per the framework of reinforcement learning, [LeggHutter2007b]); 3) the capacity to perform classification (popularly known as the VC dimension, named after their proponents Vapnik and Chervonenkis [VapnikChervonenkis2015]).

However, the above measures are impractical because: 1)  $\Phi$  is estimated from input-output relations at the component level and hence not computable for black-box systems like our nanomaterial networks; 2) the expression for universal intelligence contains the Kolmogorov complexity function [Ming1990] which is not computable; 3) the VC dimension is an all-or-nothing measure, e.g. a physical system can realize a high number of logic functions and yet have a VC dimension = 1 as shown in Figure 6.1.

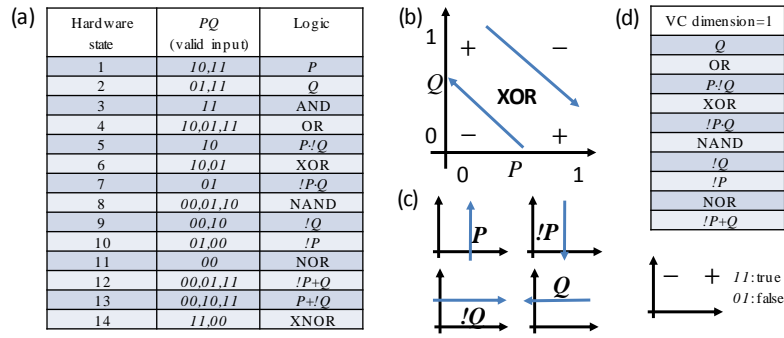


Figure 6.1. (a) Table of functions in a physical system where for each hardware state (for example, could be a set of applied voltages or interconnections), the output is true only for the valid set of input, thus realizing a unique logic (from a binary perspective). It has a VC dimension = 4, because every possible outcome of 4 points in the input space can be *shattered* as explained next. (b) The logic function XOR visualised as a shattering of the 4 points in the input space into a + region (where output is true) and a - region. (c) A minimal set of 4 functions for a VC dimension = 2. (d) Above, a set of 10 functions with a VC dimension = 1, because they cannot shatter the outcome of 2 points shown below.

In our work, we model intelligence as the capacity to decide if a *pattern* relates to a *datum* as shown in Figure 6.2a. The physical system is treated as a black-box which outputs the decision as true, if there is a relation between the datum and the pattern. This objective perspective is a generalization of the binary perspective explained in Figure 6.1, and may be implemented in our NP cluster (where the output current is a function of the input signal and voltage configuration) and NP internet (where the output current is a function of the input layer wiring and hidden layer wiring) as shown in Figure 6.2b. Thus, the input-output relations across several configurations of our physical systems can be translated into a table of relations between data and patterns as shown in Figure 6.2c, from which its IQ is to be determined. Each row of relation in this table contains a unique subset of data, and patterns that relate to all of them.

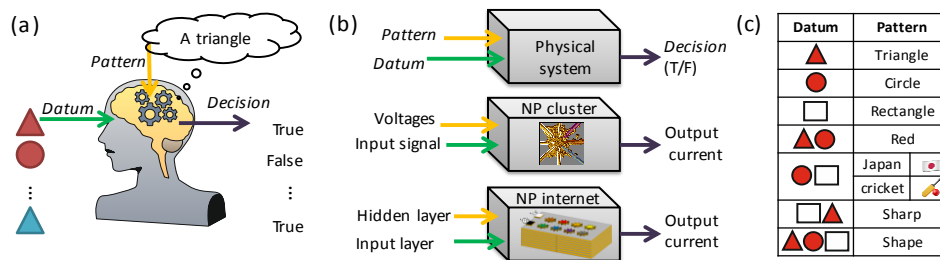


Figure 6.2. (a) Our model of an intelligent system working as a decision machine. The decision is true when there is a relation between the datum and the pattern. (b) Implementation of our model on our physical systems. The pattern and datum can be mapped to subsets of the voltage or wiring configuration. The decision is true if the output current is above a threshold. (c) An example table of 7 relations, with 3 data and 10 patterns.

### 6.1.1 IQ factors of a table of relations

In order to obtain an IQ metric from a table of relations, we can consider the following factors to be relevant:

- 1) The number of relations  $N_R$ . It is also the total number of logic functions, from a binary perspective.
- 2) The number of data  $N_D$  and patterns  $N_P$ . They are a measure of *information*.
- 3) The geometric mean of the number of data per relation  $\mu_D$ . It is a measure of *integration* (several data relating to a pattern).
- 4) The geometric mean of the number of patterns per relation  $\mu_P$ . It is a measure of *evolvability* (abundance of configurations per logic from a binary perspective).

For example, the IQ factors of the table in Figure 6.2c are  $(N_R, N_D, N_P, \mu_D, \mu_P) = (7, 3, 10, 2^{\frac{3}{7}}3^{\frac{1}{7}}, 4^{\frac{1}{7}})$ . A simple IQ metric could be the log-average of these factors, thus resulting in an IQ 
$$= \frac{\log_2 7 \times 3 \times 10 \times 2^{\frac{3}{7}} 3^{\frac{1}{7}} \times 4^{\frac{1}{7}}}{5} = 1.73 \text{ bits.}$$

In the following section, we shall apply this methodology to study the total number of logic functions present in a physical system based on experimental measurements.

## 6.2 IQ test for Boolean logic in a nanomaterial cluster

When searching for Boolean logic in our NP cluster, optimizing the voltage configurations to find different logics was a time-intensive task. If we want to repeat such experiments on novel physical systems, it is useful to gauge their intelligence in a time-efficient manner. For that purpose, we performed the following IQ test on a nanomaterial cluster of dopant atoms (lightly doped Si:B, detailed in [vanGelder2017]).

The nanomaterial cluster consists of 8 electrodes, of which an output electrode measures current flowing to the ground and each of the remaining 7 electrodes apply voltages of either 0 V or 1 V as given by a configuration number. For example,  $V_7V_6..V_1 = 0100010$  in binary corresponds to configuration 34 in decimal. For the IQ test, the output current was measured for all  $2^7 = 128$  possible voltage configurations in 4 different runs to check for reproducibility and time stability. The delay  $\tau$  between setting the voltage configuration and measuring output current was,  $\tau = 1$  s for the first 2 runs and  $\tau = 0.1$  s in the next 2 runs. The output current was found to range between -12 to 700 pA, and is plotted in Figure 6.3. The maximum standard deviation across the different runs was  $< 50$  pA.

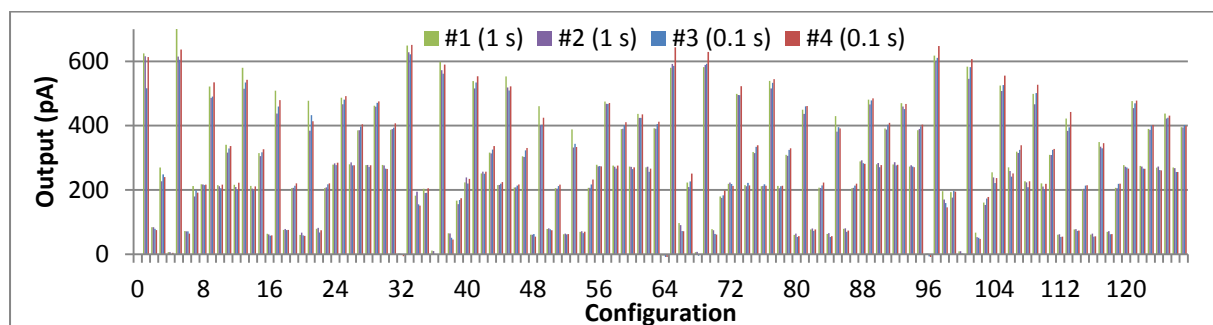


Figure 6.3. Output current vs. configuration of the nanomaterial cluster of dopant atoms. The legends correspond to 4 different runs where the output was measured after a delay of 1 s for runs #1 and #2 and 0.1 s for runs #3 and #4.

Subsets of the 7 voltage configuration bits can be mapped to a datum or pattern and a current threshold chosen in many arbitrary ways in order to obtain a table of relations similar to those shown in Figure 6.2. Here, we focus on 2-input Boolean logic where the input electrode position can be mapped to 21 different combinations (labelled from A to U in Figure 6.4a). The datum is mapped to the input voltage bits and the pattern to the combination letter followed by the decimal number due to the other 5 bits. E.g. choosing combination M with inputs  $V_5V_3$ , configuration  $34 = 0100010$  gives a datum 00 and pattern M10 (because 10 is decimal value of  $01 \cdot 0 \cdot 10$ ).

For a threshold of 100 pA, the table of relations was found to be the same for all 4 runs as shown in Figure 6.4b. In total 11 relations are present and the only missing logic are XOR, NAND, !P and XNOR. As an example, we show the output measurements corresponding to the NOR logic due

to pattern M10 in Figure 6.4c. We can see that the relation for this pattern is valid even for thresholds between ~80 to 140 pA. However, if the threshold were 160 pA, the relation for this pattern is present only for runs #1 and #2, and thus invalid. Hence, the table of relations is strongly dependent on our (arbitrary) choice of threshold currents.

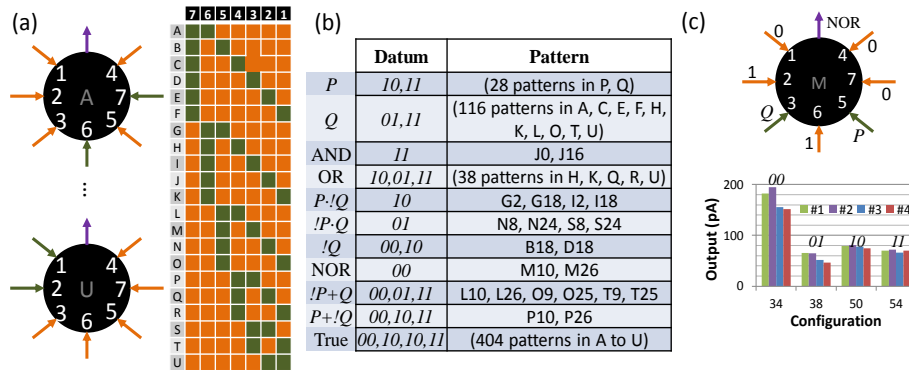


Figure 6.4. (a) All possible combinations of electrode arrangements, labelled from A to U. Input electrodes are green, control electrodes are orange and the output electrode is purple. (b) Table of relations realized for a threshold of 100 pA in the nanomaterial cluster. For some relations, since a large number of patterns were found, we only mention their total number and the set of electrode arrangements. (c) Above, is the distribution of voltages to realize NOR logic due to pattern M10. Below, is the output currents measured at configurations corresponding to each datum for the 4 different runs.

In Figure 6.5, we plot the number of relations  $N_R$  and patterns  $N_P$  for threshold values between 0 and 700 pA. Both  $N_R$  and  $N_P$  are maximum at a threshold of 100 pA. For a threshold greater than 650 pA,  $N_R$  and  $N_P$  drop to 0 because the decisions are always false. For a threshold of -50 pA,  $N_R = 1$  because the decisions are always true and thus all possible patterns ( $N_P = 21 * 2^5 = 672$ ) relate to the data (00,10,10,11).

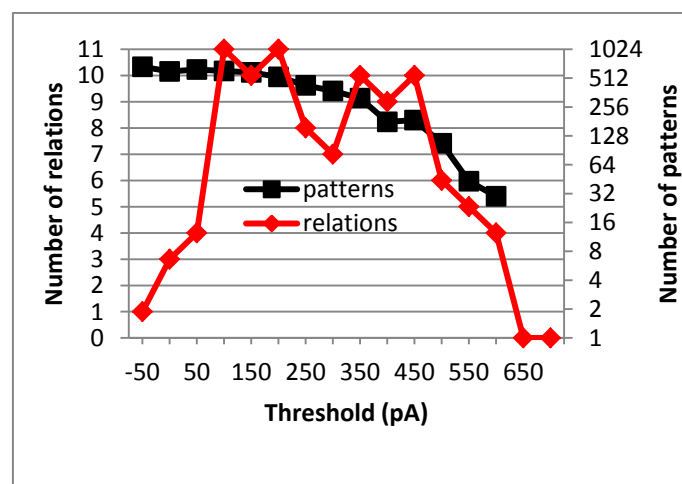


Figure 6.5.  $N_R$  and  $N_P$  vs. threshold current for measurements in the nanomaterial cluster.

## 6.3 Conclusion

We provided a definition for intelligence that is relevant for pattern recognition. As a first step, here we demonstrated the application of an IQ test for characterising the Boolean logic capacity of a nanomaterial cluster. The next step would be to apply this IQ test on simulations of SET network models of NP clusters as introduced in Chapter 5 Section 2. It is expected that physical parameters influencing the amount of crosstalk and disorder in the cluster play an important role. Optimizing physical parameters for maximal IQ could lead to a more intelligent design in electrode structures, sizes and concentration of NPs, and thickness of the molecular coatings.

It is also yet to be demonstrated how well the IQ factors of a NP cluster for Boolean logic approximate the abundance of functionality in a NP internet for pattern recognition as proposed in Chapter 5 Section 4. Developing an IQ test based on current-voltage relations of a NP internet across interesting wiring configurations is recommended. Relations of the same IQ test could also be used to estimate inhibitory functionality.

The application of our IQ metric in evaluating artificial neural networks used in deep learning and biological networks should provide more insights on the practicality of our IQ metric.

## REFERENCES

1. [LeggHutter2007] Legg, S., & Hutter, M. (2007). A collection of definitions of intelligence. *Frontiers in Artificial Intelligence and applications*, 157, 17.
2. [Oizumi2014] Oizumi, M., Albantakis, L., & Tononi, G. (2014). From the phenomenology to the mechanisms of consciousness: integrated information theory 3.0. *PLoS computational biology*, 10(5), e1003588.
3. [Tegmark2015] Tegmark, M. (2015). Consciousness as a State of Matter. *Chaos, Solitons & Fractals*, 76, 238-270.
4. [LeggHutter2007b] Legg, S., & Hutter, M. (2007). Universal intelligence: A definition of machine intelligence. *Minds and Machines*, 17(4), 391-444.
5. [VapnikChervonenkis2015] Vapnik, V. N., & Chervonenkis, A. Y. (2015). On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity* (pp. 11-30). Springer International Publishing.
6. [Ming1990] Ming, L., & Vitányi, P. M. (1990). Kolmogorov complexity and its applications. In *Algorithms and Complexity* (pp. 187-254).
7. [vanGelder2017] van Gelder, J. (2017). *Impurity Atom Network for Neuromorphic Computing* (Master's thesis). Enschede: University of Twente.

## 7 Outlook

To put the importance of our work into perspective, we shall discuss here historical and contemporary works that fall in its vicinity and provide avenues for future research.

### 7.1 Evolution in materio

The field of EIM has experiments tracing back to the 1950's, when the cyberneticist Gordon Pask manually evolved metallic threads between electrodes present in a metal salt solution to distinguish between sound frequencies, and called it an ear [Cariani1993]. Many more experiments followed [Miller2014], of which attempts to evolve the voltage configuration and connectivity in a liquid crystal board to realize Boolean logic gates deserves special mention [Harding2005]. However, evolution of Boolean logic gates such as XOR faced difficulties concerning stability and reconfigurability. It was reasoned that the liquid crystals had a form of memory in their orientation, which was hard to control.

In our work discussed in Chapter 4, we show that it is possible to evolve the complete set of 2-input Boolean logic gates in clusters of gold nanoparticles at ~300 mK. We explicitly evolve the major logic gates, and show that the AND and NAND gate are stable and reconfigurable even after a span of 100 hours and a thermal cycle of heating up to 15 K and cooling back to 300 mK. This highlights that the logic gates found by evolution are robust and immune to minor background charge fluctuations. When thermally cycling to room temperature and back, because background charges can get redistributed with much higher probability, the previously evolved logic gates did not work anymore and thus the gates had to be evolved once more. This could mean that evolution is sensitive to the spatially varying background charge and exploits them for functionality.

Following our work on NP clusters, students in our NanoElectronics group, Jeroen van Gelder [vanGelder2017] and William Elferink [Elferink2018], have shown in their Master theses that Boolean logic can be evolved in clusters of dopants (boron or arsenic atoms in a silicon chip) cooled inside a dewar of liquid helium or liquid nitrogen. Elferink used the IQ test methodology discussed in Chapter 6 to identify the best electrode assignments for evolving Boolean logic. Future experiments are planned to interconnect several clusters to form a dopant internet and evolve functionality such as digit recognition as discussed in Chapter 5. Efforts are also being made to attain functionality at room temperature, to enable integration into consumer electronics. Nathan Blanken and Laurens Westenberg investigated clusters of ionic complexes for this purpose in their BSc project [BlankenWestenberg2015]. Although nonlinearity was found, we are yet to observe SET physics as reported in single-atom transistors [Park2002].

Another promising nanomaterial for EIM is a network of carbon nanotubes (CNT), because SET physics [Postma2001] and logic circuits [Bachtold2001] have been demonstrated using them at room temperature. Mohid *et al.* [Mohid2014] attempted EIM in a network of CNT dispersed in acrylic

on top of microscale electrodes, to solve machine learning classification of datasets. They achieved an average test accuracy of ~77% on classifying the Iris dataset, which is a problem that was originally solved in [Fisher1936] by linear discriminant analysis. This means it is an easy problem that can also be solved by a purely resistive network. In contrast, our work in Chapter 4 on evolving Boolean logic gates such as XOR is a proof of much richer physics. The XOR problem is linearly inseparable and thus can not be solved by neural networks without any hidden layers. So far, attempts to evolve XOR in networks of CNT (on top of microscale electrodes) had failed [Kotsialos2014]. Future experiments to evolve functionality in networks of CNT are recommended to be on top of nanoscale electrodes (at room temperature).

As we go further down the nanoscale to achieve room temperature operation, quantum confinement effects will begin to play an important role. Decades ago, a nanomaterial cluster of polysilicon grains was demonstrated to work as a single-electron memory device at room temperature [Yano1994]. For that cluster of “quantum dots”, local variations in thickness between 5 and 2 nm result in wildly varying quantum confinement potentials between 50 and 300 meV, and thus the potential landscape (called “the nanometer Grand Canyon”) varies considerably in character across devices, providing more diversity to apply EIM. Quantum effects, such as resonant tunnelling induced NDR [Saitoh2004], enrich the physics to be exploited by EIM.

## 7.2 Evolution and the brain

Evolution of life on Earth is a slow process limited by the time to reproduce and rates of mutation. It is estimated to have taken billions of years to evolve from single-celled life to multicellular life and another 10 million years to evolve from earlier apes to modern humans. But there are also evolutionary processes in nature that can happen at much shorter time scales. A relevant example is Edelman’s theory of neuronal group selection [Edelman1993] where he argues that higher order brain functions such as the ability of organisms to categorize an unlabelled world and behave in an adaptive fashion arises not from instruction or information transfer (as in a backpropagation algorithm), but from processes of selection upon variation (of the synaptic interconnections in their brain). Beginning with the blank state (*tabula rasa*) of an embryonic brain we can evolve to the talents of an adult brain in a matter of years. Thus, we could argue that nature took billions of years to evolve organisms that could seed a brain capable of evolving its wiring configuration quickly.

Looking at functional models of brain regions [Eliasmith2012], we can see that the interconnectivity in the basal ganglia (responsible for action selection) that is modulated by a reward evaluation serves as a suitable region for evolution in the brain. But how can we reckon an evolutionary theory of brain connectivity leading to apparent functional specificity of regions and anatomical hierarchy [Felleman1991] in regions such as the visual and auditory cortex that is ‘common’ across humans? Actually functional specificity is only restricted to generic concepts like faces, places,



and bodies but no evidence of cortical specialization for concepts such as predators, prey, tools, food, cars, chairs, etc. were found [Kanwisher2010]. Studies on blind children conclude that spoken language perception colonizes the visual system during brain development [Bedny2015,Lane2015], suggesting that the human cortex may have a broad computational capacity. Thus anatomical hierarchies do not always enforce functional specificity in the brain, and the functionality is sensitive to the input during development, an emergent behaviour. This reminds us of Toffoli’s argument [Toffoli2004] that computation cannot be even properly defined if it is not placed in the context of the evolutionary feedback loops that brought it into being. Because functional models, while indispensable for understanding complex systems, are just approximations!

So there is reason to believe that evolution on neural networks can lead to human-level intelligence even if they do not map to the functional or anatomical architecture of the human brain. Moreover, bio-inspiration probably has a better chance at outdoing nature than bio-mimicry. This gives us confidence that simplistic large-scale small-world networks, such as the NP internet proposed in Chapter 5, may be enough to realize intelligence.

### **7.3 Neuroevolution in nanomaterio**

We are now at the cusp of an AI boom; several difficult problems like pattern recognition are being more efficiently solved by evolutionary algorithms on artificial neural networks. Recent growth in computing power enabled techniques like evolution strategies [Salimans2017] and deep neuroevolution [Such2017] to become a scalable alternative to backpropagation based deep learning methods for training networks with millions of weights to play video games at super-human level. The field of neuroevolution has made remarkable progress over the last two decades [Soltoggio2017], however implementations on neuromorphic hardware have not been realized yet.

For future experiments on neuroevolution, we propose that a nanomaterial system like our NP internet is much more suitable than neuromorphic hardware designed out of digital components like TrueNorth. A specific reason is the energy-efficiency plausible in our system by operating at the single-electron level. More profound reasons that are applicable to any nanomaterial system are super-Turing computation [Siegelmann1995] which is possible in analog neural networks, better signal to noise ratio due to stochastic resonance [Wiesenfeld1995], and all the other advantages of EIM.

### **7.4 Overview**

Apart from our work, other interesting nanomaterial systems proposed for neuromorphic computing are self-assembled atomic switch networks (ASN) [Demis2016,Scharnhorst2017] that serve as a reservoir for temporal dynamics and semi-predictable logic, spintronic oscillators [Torrejon2017] that serve as a reservoir to perform spoken digit recognition, and memristive crossnets [Prezioso2015] that

classify 3×3 binary input images. However, inhibitory functionality has not been demonstrated in any other nanomaterial network apart from our work on NP networks. This makes NP networks a better candidate for realizing classifiers based on non-temporal pattern recognition.

In Table 7.1 we present an overview of the natural computing systems discussed and network property requirements that we consider essential for neuroevolution in nanomaterialio. To our knowledge, the NP internet is the only system satisfying all those requirements. This is not surprising given that both EIM and neuroevolution (in software) are still emerging fields. A common theme in AI is that many great ideas were neglected earlier simply because we did not have the necessary computational resources to execute them. In that sense, our proposition for neuroevolution in nanomaterialio is timely!

Table 7.1. State-of-the-art in natural computing compared to the human brain.

Network Property	CNT	ASN	NP cluster	TrueNorth (CMOS)	Memristive CrossNet	NP internet	Human Brain
Nonlinear	✓	✓	✓	✓	✓	✓	✓
Inhibitory	×	×	✓	✓	×	✓	✓
Large-scale Small-world	×	×	×	✓	✓	✓	✓
Nanoscale	×	×	✓	✓	✓	✓	×
Designless	✓	✓	✓	×	×	✓	✓
Analog computing	✓	✓	✓	×	✓	✓	✓

## REFERENCES

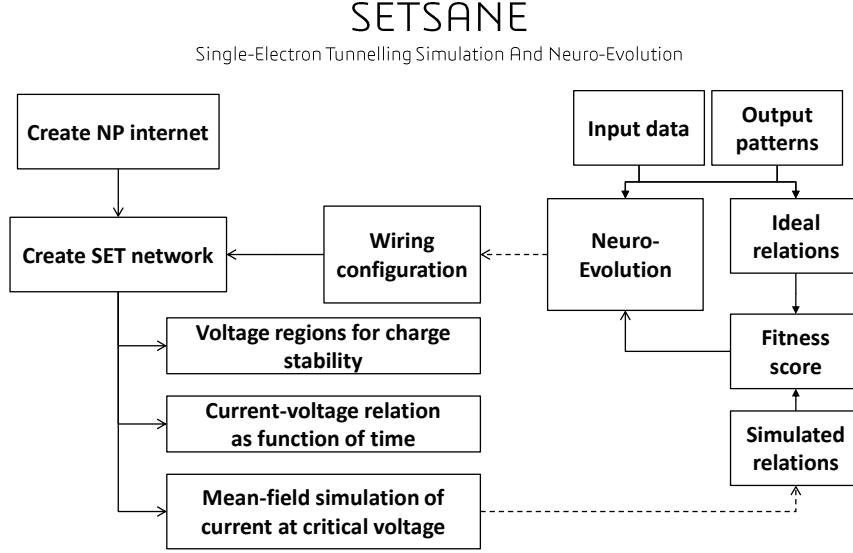
- [LeggHutter2007] Legg, S., & Hutter, M. (2007). A collection of definitions of intelligence. *Frontiers in Artificial Intelligence and applications*, 157, 17.
- [Cariani1993] Cariani, P. (1993). To evolve an ear. Epistemological implications of gordon pask's electrochemical devices. *Systems Research and Behavioral Science*, 10(3), 19-33.
- [Miller2014] Miller, J. F., Harding, S. L., & Tufte, G. (2014). Evolution-in-materio: evolving computation in materials. *Evolutionary Intelligence*, 7(1), 49-67.
- [Harding2005] Harding, S., & Miller, J. F. (2005). Evolution in materio: Evolving logic gates in liquid crystal. In *Proc. Eur. Conf. Artif. Life (ECAL 2005), Workshop on Unconventional Computing: From cellular automata to wetware* (pp. 133-149). Beckington, UK.
- [vanGelder2017] van Gelder, J. (2017) *Impurity Atom Network for Neuromorphic Computing* (Master's thesis). Enschede: University of Twente.
- [Elferink2018] Elferink, W. (2018) *Reservoir computing in disordered particle networks* (Master's thesis). Enschede: University of Twente.
- [BlankenWestenberg2015] Blanken, N., & Westenberg, L. (2015). *Scanning probe imaging of current flow in a designless network of gold nanoparticles and ferrocene dendrimers showing Coulomb blockade behaviour at room temperature* (Bachelor's thesis). Enschede: University of Twente.
- [Park2002] Park, J., Pasupathy, A. N., Goldsmith, J. I., Chang, C., Yaish, Y., Petta, J. R., ... & Ralph, D. C. (2002). Coulomb blockade and the Kondo effect in single-atom transistors. *Nature*, 417(6890), 722.
- [Postma2001] Postma, H. W. C., Teepen, T., Yao, Z., Grifoni, M., & Dekker, C. (2001). Carbon nanotube single-electron transistors at room temperature. *Science*, 293(5527), 76-79.
- [Bachtold2001] Bachtold, A., Hadley, P., Nakanishi, T., & Dekker, C. (2001). Logic circuits with carbon nanotube transistors. *Science*, 294(5545), 1317-1320.
- [Mohid2014] Mohid, M., Miller, J. F., Harding, S. L., Tufte, G., Lykkebø, O. R., Massey, M. K., & Petty, M. C. (2014, September). Evolution-in-materio: Solving machine learning classification problems using materials. In *International Conference on Parallel Problem Solving from Nature* (pp. 721-730). Springer, Cham.

12. [Fisher1936] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2), 179-188.
13. [Kotsialos2014] Kotsialos, A., Massey, M. K., Qaiser, F., Zeze, D. A., Pearson, C., & Petty, M. C. (2014). Logic gate and circuit training on randomly dispersed carbon nanotubes. *International journal of unconventional computing.*, 10(5-6), 473-497.
14. [Yano1994] Yano, K., Ishii, T., Hashimoto, T., Kobayashi, T., Murai, F., & Seki, K. (1994). Room-temperature single-electron memory. *IEEE Transactions on Electron Devices*, 41(9), 1628-1638.
15. [Saitoh2004] Saitoh, M., & Hiramoto, T. (2004). Extension of Coulomb blockade region by quantum confinement in the ultrasmall silicon dot in a single-hole transistor at room temperature. *Applied physics letters*, 84(16), 3172-3174.
16. [Edelman1993] Edelman, G. M. (1993). Neural Darwinism: selection and reentrant signaling in higher brain function. *Neuron*, 10(2), 115-125.
17. [Eliasmith2012] Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., & Rasmussen, D. (2012). A large-scale model of the functioning brain. *science*, 338(6111), 1202-1205.
18. [Felleman1991] Felleman, D. J., & Van, D. E. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1), 1-47.
19. [Kanwisher2010] Kanwisher, N. (2010). Functional specificity in the human brain: a window into the functional architecture of the mind. *Proceedings of the National Academy of Sciences*, 107(25), 11163-11170.
20. [Bedny2015] Bedny, M., Richardson, H., & Saxe, R. (2015). "Visual" cortex responds to spoken language in blind children. *Journal of Neuroscience*, 35(33), 11674-11681.
21. [Lane2015] Lane, C., Kanjlia, S., Omaki, A., & Bedny, M. (2015). "Visual" cortex of congenitally blind adults responds to syntactic movement. *Journal of Neuroscience*, 35(37), 12859-12868.
22. [Toffoli2004] Toffoli, T. (2004). Nothing Makes Sense in Computing Except in the Light of Evolution. *IJUC*, 1(1), 3-29.
23. [Salimans2017] Salimans, T., Ho, J., Chen, X., & Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*.
24. [Such2017] Such, F. P., Madhavan, V., Conti, E., Lehman, J., Stanley, K. O., & Clune, J. (2017). Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning. *arXiv preprint arXiv:1712.06567*.
25. [Soltoggio2017] Soltoggio, A., Stanley, K. O., & Risi, S. (2017). Born to Learn: the Inspiration, Progress, and Future of Evolved Plastic Artificial Neural Networks. *arXiv preprint arXiv:1703.10371*.
26. [Siegelmann1995] Siegelmann, H. T. (1995). Computation beyond the Turing limit. *Science*, 268(5210), 545-548.
27. [Wiesenfeld1995] Wiesenfeld, K., & Moss, F. (1995). Stochastic resonance and the benefits of noise: from ice ages to crayfish and SQUIDS. *Nature*, 373(6509), 33.
28. [Demis2016] Demis, E. C., Aguilera, R., Scharnhorst, K., Aono, M., Stieg, A. Z., & Gimzewski, J. K. (2016). Nanoarchitectonic atomic switch networks for unconventional computing. *Japanese Journal of Applied Physics*, 55(11), 1102B2.
29. [Scharnhorst2017] Scharnhorst, K., Woods, W., Teuscher, C., Stieg, A., & Gimzewski, J. (2017, July). Non-temporal logic performance of an atomic switch network. In *2017 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)* (pp. 133-138). IEEE.
30. [Torrejon2017] Torrejon, J., Riou, M., Araujo, F. A., Tsunegi, S., Khalsa, G., Querlioz, D., ... & Kubota, H. (2017). Neuromorphic computing with nanoscale spintronic oscillators. *Nature*, 547(7664), 428.
31. [Prezioso2015] Prezioso, M., Merrih-Bayat, F., Hoskins, B. D., Adam, G. C., Likharev, K. K., & Strukov, D. B. (2015). Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature*, 521(7550), 61.

## A Appendix

### 1) Mathematica code

Mathematica code for MF simulation of SET networks, stability region plots and simulation of NP internets can be found in <https://github.com/celestine-preetham/SETSANE>.



### 2) Numerical method to solve for equilibrium point of MF approximation

AIM: To find the equilibrium point  $\mathbf{n}$  such that  $\frac{d\mathbf{n}}{dt} = \frac{I_B(\mathbf{n}) + I_O(\mathbf{n}) + I(\mathbf{n})}{e} = \mathbf{0}$

OVERVIEW: A standard method of explicit time integration (or implicit in case of stiff systems) is computationally overkill when transients are of no interest. Alternatively, multidimensional root-finding is tricky as 1) in derivative based methods computing the Jacobian matrix is expensive for large dimensions and confusing in presence of multiple local minima 2) bracketing methods may be impossible when the function is non-convex. So, here we introduce a *confidence controlled multidimensional stepping* method.

INPUT: functions  $I_B(\mathbf{n}), I_O(\mathbf{n}), I(\mathbf{n})$ , initial state  $\mathbf{n}_0 = \mathbf{0}$ , initial step-confidence  $\delta = \text{Normalize} \left[ \frac{d\mathbf{n}}{dt} \right]$ , gain factor  $\gamma = 1.2$ , loss factor  $\lambda = 0.5$ , tolerance=0.01.

RUN:  $step=0$ ;  $\mathbf{n} = \mathbf{n}_0$ ; **Do** TakeStep;  $step++$  **While**  $step < 200$  &&  $precision < \text{tolerance}$ .

TakeStep:= {

$\Delta \mathbf{n} = \frac{I_B(\mathbf{n}) + I_O(\mathbf{n}) + I(\mathbf{n})}{e}$ ; (compute rate of change)

**If**  $|\Delta \mathbf{n}| < 10^{-7}$ , **Then Stop**;

**For**  $[r = 1 \text{ to } N, (\text{for each dimension})$

**If**  $\Delta n_r \delta_r \geq 0$  (check if rate of change and step-confidence have the same sign)

**Then**  $\delta_r = 10^{-8} \text{Sign}[\Delta n_r] + \text{Clip}[\gamma \delta_r, \pm 0.1]$  (step forward with a gain in confidence, with magnitude clipped between  $10^{-8}$  and 0.1)

**Else**  $\delta_r = -\lambda \delta_r$  (step backward with a loss in confidence)

];

$\mathbf{n} = \mathbf{n} + \delta$  (update state);  $\text{precision} = \text{RelativeError}[\Sigma I_O(\mathbf{n}), -\Sigma I_B(\mathbf{n})]$  }.

### 3) Relative error

$\text{RelativeError}[I_1, I_2] := \frac{|I_1 - I_2|}{|I_1| + |I_2| + I_n}$  (The noise current  $I_n = 10^{-13}$  A is added so that the absolute error is  $|I_1 - I_2| < \text{tolerance} \times I_n = 1$  fA for a tolerance=1%).

### 4) Netlist of disordered network

Our SET network consists of  $\alpha = 1$  to  $N$  particles (of radius  $r_p = 10$  nm) that are dispersed randomly at positions  $\mathbf{r}(\alpha)$  in a circle of radius 100 nm between electrodes at positions  $\mathbf{r}(B) = (10r_p, 0)$  and  $\mathbf{r}(O) = (-10r_p, 0)$ . We consider a characteristic capacitance  $C_0 = 10e = 1.6$  aF, characteristic resistance  $R_0 = 10^5$  Ohm and leakage resistance  $R_\infty = 10^{16}$  Ohm. We assume that  $C_{G\alpha} = C_0(1 + \text{Random}[0,1])$ ,  $R_{G\alpha} = R_\infty$  for  $\alpha = 1$  to  $N$  and  $C_{\alpha\beta} = C_0 r_p / d(\alpha, \beta)$ ,  $R_{\alpha\beta} = R_0 E^t(\alpha, \beta)$  where  $d(\alpha, \beta) = |r_p + r(\alpha) - r(\beta)|$ ,  $t(\alpha, \beta) = \left| \frac{r(\alpha) - r(\beta)}{r_p} \right|$  for  $\alpha, \beta \in \{B, O, 1, \dots, N\}$ .

The netlist for the 2-particle disordered network is

```
TEMPERATURE= 0.000000e+00
q-MQT_ORDER= 1
SEED=-2
MEASUREMENT_MODE=TRANSIENT
START_TIME= 0.000000e+00
END_TIME= .99
TIME_STEP= 1.000000e-02
EVENT_NR_LIMIT= 1700
MIN_STATE_PROB= 1.000000e-10
MAX_STATE_PROB_ERR= 1.000000e-03
LIMIT_RATE_FRACTION= 1.000000e+01
MAX_LEVEL= 5
J5 1 2 C=1.118890e-19 R="4.353520e8" NV1=-1 NV2=-1 SD=TRUE {35, 70, 0, STRAIGHT,STRAIGHT}
J7 1 3 C=1.867430e-18 R="1.010e16" NV1=-1 NV2=-1 SD=TRUE {35, 105, 0, STRAIGHT,STRAIGHT}
J9 1 4 C=8.801250e-20 R="2.369840e9" NV1=-1 NV2=-1 SD=TRUE {35, 140, 0, STRAIGHT,STRAIGHT}
J11 1 5 C=7.285370e-20 R="1.490460e11" NV1=-1 NV2=-1 SD=TRUE {35, 175, 0, STRAIGHT,STRAIGHT}
J8 2 3 C=1.31630e-18 R="1.010e16" NV1=-1 NV2=-1 SD=TRUE {70, 105, 0, STRAIGHT,STRAIGHT}
J10 2 4 C=8.743190e-20 R="4.074650e9" NV1=-1 NV2=-1 SD=TRUE {70, 140, 0, STRAIGHT,STRAIGHT}
J12 2 5 C=1.195470e-19 R="1.360250e9" NV1=-1 NV2=-1 SD=TRUE {70, 175, 0, STRAIGHT,STRAIGHT}
V1 3 0 V=0.0e0 {105, 25, 180,1,STRAIGHT,STRAIGHT}
V2 4 0 V=0.0e0 {140, 25, 180,1,STRAIGHT,STRAIGHT}
V3 5 0 V=9.820030e-2 {175, 25, 180,1,STRAIGHT,STRAIGHT}
N 1 Q=0.0e0 VOLATILE=ON MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00 {35, 35}
N 2 Q=0.0e0 VOLATILE=ON MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00 {70, 70}
N 3 Q=0.000000e+00 VOLATILE=ON MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00 {105, 105}
N 4 Q=0.000000e+00 VOLATILE=ON MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00 {140, 140}
N 5 Q=0.000000e+00 VOLATILE=ON MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00 {175, 175}
.Q1 1 {35, 35, 0,STRAIGHT}
.Q2 2 {70, 70, 0,STRAIGHT}
Z 0 {10,10,1}
ZERO_NODE_INFO MATERIAL:METAL EC=0.000000e+00 EV=0.000000e+00
.I1 V2 ORIENTATION=POSITIVE {0,0,0,STRAIGHT}
.I2 V3 ORIENTATION=POSITIVE {0,0,0,STRAIGHT}
```

### 5) Netlist of 4×4 grid network

All the tunnel junctions for adjacent locations  $\alpha, \beta$  on the grid have  $C_{\alpha\beta} = C_0 = 1.6 \times 10^{-18}$  aF and.  $R_{\alpha\beta} = 10^5 \Omega$ . The islands have a gate capacitance  $C_{G\alpha} = 2C_0$ . The input voltage amplitude is 10 mV and the control voltages (in V) are given by the gene values below.

	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$b$
XOR	0.00458	0.00466	0.0108	0.0212	-0.0022	0.338
AND	0.011	0.0021	0.00941	-0.00124	0.00454	-0.131
OR	-0.00446	-0.0128	0.00748	-0.00202	-0.00791	-0.676
NAND	0.0156	0.0215	0.0082	0.0161	0.0221	-0.014

### 6) Netlist of bistable circuit

$\{C_{xz}, C_{x0}, C_{y0}, C_{xg}, C_{yz}, C_{xy}, R_{xz}, R_{x0}, R_{yz}, C_{zs}, R_{zs}\} = \{100, 100, 101, 1, 100, 300, 1, 1, 1, 500, 1\}$ ;  $v_g = 0.08$  V.

### 7) SIMON netlist of multiscale network

```

TEMPERATURE= 0.000000e+00
q-MQT_ORDER= 1
SEED= -1
MEASUREMENT_MODE=QUASI_STATIONARY
START_TIME= 0.000000e+00
END_TIME= 1.000000e+00
TIME_STEP= 1.000000e-02
EVENT_NR_LIMIT= 10000
MIN_STATE_PROB= 1.000000e-10
MAX_STATE_PROB_ERR= 1.000000e-03
LIMIT_RATE_FRACTION= 1.000000e+01
MAX_LEVEL= 5
C1 6 0 C=1.000000e-15 {286,386,180,5,STRAIGHT,STRAIGHT}
C2 0 5 C=3.000000e-18 {286,469,0,6,STRAIGHT,STRAIGHT}
C3 4 0 C=3.000000e-18 {287,297,0,4,STRAIGHT,STRAIGHT}
C4 13 0 C=3.000000e-18 {414,366,0,10,STRAIGHT,STRAIGHT}
C5 12 0 C=3.000000e-18 {204,356,0,2,STRAIGHT,STRAIGHT}
J1 0 13 C=1.000000e-18 R="1.000e+05" NV1=-1 NV2=-1 SD=FALSE {463,364,180,11,STRAIGHT,STRAIGHT}
J2 4 13 C=1.000000e-18 R="1.000e+05" NV1=-1 NV2=-1 SD=FALSE {334,297,0,STRAIGHT,STRAIGHT}
J3 3 12 C=1.000000e-18 R="1.000e+05" NV1=-1 NV2=-1 SD=FALSE {153,355,0,STRAIGHT,STRAIGHT}
J4 5 13 C=1.000000e-18 R="1.000e+05" NV1=-1 NV2=-1 SD=FALSE {343,469,0,STRAIGHT,STRAIGHT}
J5 6 5 C=1.000000e-18 R="1.000e+05" NV1=-1 NV2=-1 SD=FALSE {286,429,90,STRAIGHT,STRAIGHT}
J6 12 4 C=1.000000e-18 R="1.000e+05" NV1=-1 NV2=-1 SD=TRUE {248,298,0,STRAIGHT,STRAIGHT}
J7 12 5 C=1.000000e-18 R="1.000e+05" NV1=-1 NV2=-1 SD=TRUE {247,469,0,STRAIGHT,STRAIGHT}
J8 4 6 C=1.000000e-18 R="1.000e+05" NV1=-1 NV2=-1 SD=TRUE {286,352,90,STRAIGHT,STRAIGHT}
V1 3 0 PWL: 0.000000e+00 0.000000e+00 1.000000e+00 4.000000e-02 {104,399,0,12,STRAIGHT,STRAIGHT}
.U1 3 0 {56,353,90,12,STRAIGHT,STRAIGHT}
.Q1 4 {346,341,0,STRAIGHT}
.Q2 6 {348,394,0,STRAIGHT}
.I1 J1 ORIENTATION=NEGATIVE {548,343,0,STRAIGHT}
.SD1 CONTROL=C1 0.000e+00 1.000e-16 100 CONTROL=V1 0.000e+00 4.000e-02 100 100 {436,438}
N 3 Q=0.000000e+00 VOLATILE=ON MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00 {118,368}
N 4 Q=0.000000e+00 VOLATILE=ON MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00 {287,298}
N 5 Q=0.000000e+00 VOLATILE=ON MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00 {286,469}
N 6 Q=0.000000e+00 VOLATILE=ON MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00 {286,387}
N 12 Q=0.000000e+00 VOLATILE=ON MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00 {204,355}
N 13 Q=0.000000e+00 VOLATILE=ON MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00 {413,365}
Z 0 {192,357,2}
Z 0 {275,298,4}
Z 0 {273,387,5}
Z 0 {274,470,6}
Z 0 {402,366,10}
Z 0 {527,399,11}
Z 0 {105,443,12}
ZERO_NODE_INFO MATERIAL: METAL EC=0.000000e+00 EV=0.000000e+00
T "C1" {302,366}

```

## 8) Fabrication details

The samples shown in Chapter 4, Figure 2a-c were fabricated by Saurabh Bose with Au electrodes. The sample shown in Chapter 4, Figure 2d was fabricated by Reinier Nielen with Pd electrodes using the following procedure

- PMMA A2 spin at 2000 rpm for ~60 seconds
- Bake at 160 degree Celsius for 120 seconds
- Write the EBL pattern using 20kV, 20  $\mu\text{m}$  aperture with the parameters tabulated below

Table A.1. Electron beam lithography parameters.

Beam current = 0.107 nA	Area	Curved Elements	Lines	Dots
Step size ( $\mu\text{m}$ )	0.01	0.01	0.0020	
Dwell time (ms)	1.4e-3	1.4e-3	0.934e-3	0.09
Dose ( $\mu\text{C}/\text{cm}^2$ )	150	150	500e-9	0.01e-9

- Develop the resist with MIBK:IPA for 33 seconds
- Rinse with isopropanol for 30 seconds
- Deposit metal by EBPVD with the parameters tabulated below

Table A.2. Electron-beam physical vapor deposition parameters.

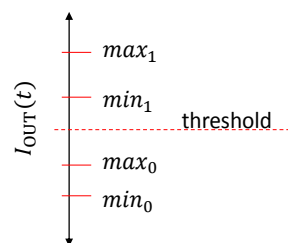
Layer #	Material	Thickness (nm)	Emission current (mA)	Rate (nm/s)	Process pressure (mbar)
1	Ti	2	36	0.03	3.0e-7
2	Pd	25	100	0.15	6.0e-7

- Lift-off with DMSO for 30 minutes in an ultrasonic bath at 80 degree Celsius
- Rinse with isopropanol and blow dry with nitrogen.

## 9) Fitness quality

Here we show how to compute the fitness quality  $\bar{F}$  of the measured output  $I_{\text{OUT}}(t)$  to satisfy a Boolean function  $f(t)$ . The bar in  $\bar{F}$  symbolizes that we use a strict definition, where  $\bar{F} = 1$  if and only if  $I_{\text{OUT}}(t) \equiv f(t)$ .

Let us define  $\min_0$  as the minimum of  $I_{\text{OUT}}(t)$  in the time windows when  $f(t) = 0$ . Similarly, we can define  $\max_0, \min_1$  and  $\max_1$ . Any threshold current between  $\min_1$  and  $\max_0$  would make  $I_{\text{OUT}}(t) \equiv f(t)$  as visualized below:



If  $\min_1 < \max_0$ , then  $\bar{F} = 0$ . because such a threshold is impossible. For a good fitness quality, we want the tolerance  $\min_1 - \max_0$  as large as possible. And so one can define  $\bar{F} = \frac{\min_1 - \max_0}{\max_1 - \min_0}$  where the tolerance is divided by the range  $\max_1 - \min_0$ , so that  $\bar{F} \leq 1$ . That is the fitness quality definition we use for memory functions. However, such a definition doesn't enforce that the offset  $\min_0$  is close to 0. For a good fitness quality in logic functions, since a low offset is desired, we define  $\bar{F} = \frac{\min_1 - \max_0}{\max_1 - \min_0 + |\min_0|}$  where  $|\min_0|$  is a penalty in the denominator.



## Acknowledgements

Here we are at the end of this booklet, which documents the work I was fortunate to partake at the University of Twente (in the peaceful little city of Enschede). “High Tech, Human Touch” is a motto which I learned here that continues to serve as a guiding principle. I would now like to express my gratitude to the people whose touch has in different ways contributed to this thesis.

I would like to start by thanking my supervisors *Wilfred* and *Hajo*. The more I researched, the more I felt double lucky to have received such a wonderful topic to work on. Thank you both for your collaboration and futuristic vision.

*Wilfred*, you continue to inspire me through your systematic way of practicing science and happy way of managing a team. Many questions you raised during our group meetings and email exchanges are attempted in this thesis. Your detailed suggestions on my writing helped a lot in improving the presentation of this thesis. And over the past 4 years, you brought in a dozen of students to work in the “Darwin on a chip” team, they have helped a lot in adding substance to this thesis. Also thanks for putting me in touch with a diverse set of professors. And lastly, thank you for your confidence in me and arranging a PhD position. Your work ethic was like a tropical sunshine!

*Hajo*, it was because of your encouragement that I had the opportunity to be in touch with the international unconventional computing community. Thank you for introducing me to blue-sky science through our NASCENCE project meeting in Trondheim, TRUCE summer school in Malaga and international conferences in Auckland and Vancouver. Also thanks for you and *Ruud* being cool profs and extending the ‘blue’s lesson further by taking me to the pop and rock museum in Trondheim. At work, you made *Dilu* wonder why the time ratio of ‘you in my office’ to ‘me in your office’ was tending to  $\infty$ . The best example is when I sent you an 11am email to discuss a graph colouring problem, you made a walk from your building to my office around noon to understand the problem, and then came back in the afternoon with a solution. Thank you for your humbleness and experience!

I would like to thank all the professors, technicians, post-docs, students and secretaries belonging to the NanoElectronics group who made it a great place to work at. *Saurabh*, thanks for being the guru of my experimental work during the first year. I still refer back to the lab book you maintained during our times in the Heliox. *Thijs*, thanks for providing all the wisdom necessary whenever it was time to design a new experimental setup and also ensuring that our lab is safe from a nanoparticle contamination or a liquid Helium explosion. *Martin*, thank you for your EE skills, they are taking this project forward. *Tao*, it was a pleasure working with you and I wish you success towards higher-temperature functionality. And last, but not the least, I would like to thank all the Bachelor and Master students who worked in our team. I am happy that, your names and references to your reports, are able to decorate every chapter after the introduction.

Special thanks to professors *Bernard Geurts*, *Peter Bobbert* and *Peter Hadley* for giving detailed comments that helped improve this thesis.

Outside academics, I would like to thank my table tennis group *Thibats* (Go Thibby!). Not only for the sports activities, but also for the *#weekendjeweg*, birthday parties, and bar duties. Special thanks to my boardgamegeek squad- ‘buddy’ *Marc*, *Wesley*, and *Chantal* who gave me lifelong memories. *Wesley*, we made the best lasagnas, and thank you for being my partner in mischief (shhh). Now, mischief reminds me of my housemates in the ‘Green light district’. You gave me a nice appreciation for Dutch student house culture *#kookrooster* *#corveerooster* *#Sinterklaas* *#kerstdiner* *#huisavond* *#huisweekend* *#huisopdracht*. You also gave me a Dutch alter ego, *Aladin*. I would miss *Aladin’s* *#jaarsen* spirit.

Lastly, I would like to thank my family for keeping me in their prayers.

## Samenvatting

De eerste grootschalige elektronische computer werd gebouwd tijdens de Tweede Wereldoorlog. Genaamd ENIAC (*electronic numerical integrator and computer*), de machine was zo groot als een stapel van 100 mensen. Het werd oorspronkelijk ontworpen om schiettafels voor artillerie te berekenen, een taak waar mensen slecht in waren. Gelukkig is de oorlog beëindigd en hebben volgende generaties computers veel aspecten van menselijke activiteit verbeterd en ons geholpen om anders onmogelijke taken zoals ruimteverkenning te bereiken. Moderne computers zijn gekrompen tot microprocessors die kleiner zijn dan een duim en worden dus veel sneller en energiezuiniger. Deze technologische vooruitgang heeft computers gemaakt die in staat zijn om applicaties zoals Microsoft's *Seeing AI* te leveren, die het landschap op de camera kunnen beschrijven om de levens van blinden te verrijken. Dit soort kunstmatige intelligentie wordt mogelijk gemaakt door *deep learning* op kunstmatige neurale netwerken (dit zijn wiskundige modellen van computernetwerken die vergelijkbaar zijn met de hersenen). *Deep learning* machines hebben recentelijk bovenmenselijke vaardigheden bereikt bij het spelen van strategische bordspellen en videogames, en hebben nieuwe kennis en intuïtie gebracht om deze spellen te spelen. We zijn echter nog steeds een maanschoot verwijderd van het bereiken van wat bekend staat als kunstmatige algemene intelligentie, die ook empathie en wijsheid omvat om beslissingen te nemen tussen goed en kwaad in de echte wereld.

De uitdaging is om neuromorfische hardware te maken die net zo grootschalig is als het menselijk brein. IBM's *TrueNorth*, de grootste neuromorfische chip sinds de afgelopen vier jaar, heeft slechts 0,01% van het aantal neuronen in een menselijk brein, geïmplementeerd op een duimformaat chip die vijf miljarden transistoren verpakt. Het verkleinen van de kenmerkgroottes naar de nanoschaal is noodzakelijk om een hogere computationele dichtheid te bereiken. Het leidt echter tot imperfecties en andere onbedoelde fysieke effecten, dus buiten het bereik van het menselijk ontwerp. Om dit op te lossen, kijken we terug naar de wegen van de natuur!

Natuurlijke computersystemen zijn een product van evolutie. Evolutie maakt gebruik van alle exploitatieerbare fysieke processen. Dus in plaats van elektronische schakelingen te maken die ontwerpregels hebben om fysieke processen zoals capacatieve overspraak uit te sluiten, lijkt het meer natuurlijk om materie in zijn ontwerploze vorm te gebruiken en deze te evolueren voor berekening. Hier laten we zien dat de elektronische eigenschappen van nanomateriaalclusters kunnen worden ontwikkeld om Booleaanse logische poorten te realiseren. Het cluster is ontwerploos in de zin dat het een ongeordende verzameling deeltjes is. Clusters van nanodeeltjes worden getest bij een temperatuur die 10 keer kouder is dan de ruimte en er wordt ontdekt dat ons systeem voldoet aan de essentiële criteria voor de kunstmatige realisatie van neurale netwerken. Evenzo functioneren clusters van atomen gedoteerd in silicium bij de temperatuur van vloeistof-stikstof, en worden prospectieve nanomaterialen voor kamertemperatuur-functionaliiteit gegeven.

Voor functionaliteit zoals beeldherkenning ontwerpen we een grootschalige architectuur met 'kleine wereld netwerk' die verschillende nanomateriaalclusters met elkaar verbindt door elektronische schakelaars waarvan de configuratie door een computerserver moet worden ontwikkeld. Door het nanomaterialen systeem te modelleren als een *single-electron tunnelling* netwerk, vinden we in simulaties dat het systeem evolueerbaar is om inhibitie te realiseren, een belangrijke neurale netwerkeigenschap. Voor simulaties van grootschalige netwerken ontwikkelen we een nieuwe gemiddelde veldbenadering en vergelijken we de nauwkeurigheid met een standaard Monte Carlo-methode. Onze gemiddelde veldmethode is meestal 10 - 100 keer sneller voor het simuleren van de stroom-spanningsverhouding.

In onze zoektocht naar het realiseren van intelligentie, is het nuttig om een wiskundige definitie en een praktische maatregel te hebben. Daartoe modelleren we intelligentie als het vermogen om patronen te relateren aan gegevens. We brengen input-output relaties van een fysiek systeem in kaart met een tabel met referentie-patroon relaties, van waaruit we een 'IQ' metriek berekenen.

Als we nadenken over recente trends in computers, zijn we getuige van een ontwerploze software-revolutie met de toenemende toepassingen van kunstmatige neurale netwerken. Het is dus logisch dat een designloze hardware-revolutie volgt en hopelijk de geboorte geeft aan een ENTHIRAN (*evolved network that has intelligence realized at nanoscale*) •

## தொகுப்பு

பேரளவு கொண்ட மின்னணு கணினி முதன்முறையாக இரண்டாம் உலகப் போரின் காலத்தில் கட்டப்பட்டது. ENIAC: electronic numerical integrator and computer என்ற அந்த இயந்திரத்தின் பரிமாணம் 100 மனிதர்களின் அடுக்கு போன்றது. அதன் ஆரம்ப நோக்கம் பீரங்கி துப்பாக்கிச்சூட்டுக்கான அட்டவணையை சீக்கிரமாய் கணக்கிடுவதற்கு, ஏனென்றால் அந்த பணிக்கு மனிதர்கள் பொருந்தாதவர்கள். நல்லவேளை போர் முடிந்தது, அதற்கப்புறம் அடுத்தடுத்தபடியாக முன்னேறிய கணினிகள் வாழ்க்கையின் பல அம்சங்களை மேம்படுத்தின மேலும் விண்வெளி ஆய்வு போன்ற மற்றபடி சாத்தியமற்ற பணிகளைச் செய்ய உதவின. நவீன கணினிகள் ஒரு கட்டைவிரல்லை விட சிறியதாக இருக்கும் நுண்செயலிகள், அதனால் அவை மிகவும் சீக்கிரமாய் மற்றும் மின்சார சிக்கனமாய் கணிப்பொறிகின்றன. இந்த தொழில்நுட்ப வளர்ச்சியால் கணினிகள் Microsoft இன் Seeing AI போன்ற பயன்பாடுகள் மூலம் ஒளிப்படக்கருவியின் காட்சியை குருடர்களுக்கு விளக்கின்றன. அவ்வகை செயற்கை நுண்ணறிவு (AI: artificial intelligence) மூளையின் கணித மாதிரியான செயற்கை நரம்பணுப் பிணையம் (artificial neural network) பயன்படுத்தி மற்றும் deep learning என்கின்ற படிமுறைத் தீர்வால் கற்கின்றது. Deep learning இயந்திரங்கள் சமீபத்தில் பலகை விளையாட்டுகள் மற்றும் videogames விளையாடுவதில் அதிபுத்திசாலி திறன்களை அடைந்தன, மேலும் அவை மனிதர்களுக்கு விளையாட புதிய அறிவும் உள்ளுணர்வுகளும் காட்டின. இருப்பினும், உலகில் நல்லது கெட்டது இடையே முடிவுகளை எடுப்பதற்கு சக உணர்வு மற்றும் ஞானம் ஆகியவற்றைக் கொண்டிருக்கும் செயற்கை பொது நுண்ணறிவு (artificial general intelligence) இன்றும் ஒரு நிலா வீச்சின் தூரம்.

இதற்கான முக்கியமான சவால், மனித மூளைக்கு ஈடான ஒரு பேரளவு கொண்ட வன்பொருள் (neuromorphic hardware) செய்வதில். கடந்த நான்கு ஆண்டாக சிறந்த neuromorphic chip ஆக இருக்கும் IBM இன் TrueNorth மனித மூளையவிட நரம்பணு அளவில் பத்தாயிரம் மடங்கு குறைந்தது. TrueNorth ஒரு கட்டைவிரல் பரப்பளவில் ஐநூறு கோடி ஆக்கக் கூறுகள் வைத்து வடிவமைக்கப்பட்டுள்ளது. அதே பரப்பளவில் நரம்பணு அளவை பெருக்க கூறுகளை சுருக்கி ஆக்க முயற்சிதால் அபூரணம் மற்றும் திட்டமிடப்படாத இயற்பியல் விளைவுகள் வந்து வடிவமைப்பை சாத்தியமில்லாக்கும். இதற்கொரு தீர்வு காண இயற்கையின் வழிகளை நினைவு காண்போம்!

இயற்கையில் கணிப்பொறிகின்ற ஒருங்கியங்கள் பரிணாம வளர்ச்சி மூலம் உருவாகின. பரிணாமம் எல்லா வித இயல்பியல் விளைவுகளிலும் லாபம் தேடும் திறன் கொண்டது. எனவே, மின்னேக்கி குறுக்குப்பேச்சு போன்ற இயற்பியல் விளைவுகளை விலக்க விதிகள் கொண்டு மின் சுற்றுவழிகள் வடிவமைப்பவதற்குப் பதிலாக, வடிவமைக்கப்படாத பொருள்களில் செயற்கை பரிணாமம் (evolution in materio) மூலம் கணிப்பொறி தயாரிக்கலாம். இங்கே, நாம் மீநுண் பொருள் (nanomaterial) கொத்துகளின் மின்னணு பண்புகளை பரிணாமம் மூலம் வளர்த்து தர்க்கவாயில்கள் (Boolean logic gates) தயாரித்து காட்டுகிறோம். கொத்து வடிவமைக்கப்படாதது என்றால் துகள்கள் ஒரு ஒழுங்கில் அடுக்காததை குறிப்பிடுது. மீநுண் துகள் கொத்துகள் விண்வெளியில் விட பத்து மடங்கு குறைந்த வெப்பநிலையில் பரிசோதிக்கப்பட்டன, மேலும் இந்த ஒருங்கியங்கள் ஒரு செயற்கை நரம்பணுப் பிணையம் உருவாக்கிட அடிப்படை தேவைகளை திருப்திபடுத்துகின்றன என்று அறிகிறோம். இதேபோல், அணு கொத்துகள் இன்னும் அதிக வெப்பநிலை கொண்ட Nitrogen திரவத்தில் செயல்படுகின்றன. அறை வெப்பநிலை செயல்பாட்டிற்கான வாய்ப்புக்கொண்ட மீநுண் பொருட்கள் வழங்கப்படுகின்றன.

பட உணர்வு (image recognition) போன்ற செயல்பாட்டிற்காக பேரளவு கொண்டிருக்கும் சிறு உலக பிணையம் (small world network) வடிவமைக்கிறோம், இந்த கட்டமைப்பில் பல்வேறு மீநுண் பொருள் கொத்துக்களை கணினி சேவையால் அமைக்கப்படுகின்ற மின் நிலைமாற்றிகள் இணைக்கின்றன. மீநுண் பொருட்கள் கொண்ட ஒருங்கியத்தை ஒரு துகள் தாவும் மின்னணு பிணையம் (single-electron tunnelling network) மாதிரி கணினி மூலம் பாவனை (simulation) செய்து ஒடைத்தடுப்பு (inhibition) என்ற முக்கியமான நரம்பணுப் பிணையம் குணத்தை பரிணாமம் மூலம் கொண்டுவர சாத்தியம் என்று அறிகிறோம். பேரளவு கொண்ட துகள் தாவும் மின்னணு பிணையங்களின் பாவனைக்காக சராசரிப் புலம் அண்ணளவாக்கம் (mean-field approximation) படைத்திருக்கிறோம். இது மின்னோட்ட-மின்னழுத்த சம்பந்தங்களை (current-voltage relations), நுண்ணியமாக பின்பற்றுகிற இயைபில்லா நிகழ்களின் பாவனை முறை: Monte Carlo simulation'யை விட 10-100 மடங்கு விரைவாக கணக்கிடுது.

நுண்ணறிவை தேடும் நம் ஆராய்ச்சிக்கு, ஒரு கணித மாதிரி மூலம் ஒருங்கியங்களை மதிப்பீடுகிற நுண்ணறிவு எண் (IQ) பயன்படும். அதற்காக நுண்ணறிவை தரவுகள் (data) உடன் தோரணிகள் (patterns) சம்பந்தப்படுத்தும் கொள்ளளவு என்று கருதுகிறோம். ஒருங்கியத்தின் உள்ளீடு-வெளியீடு சம்பந்தங்களை தரவு-தோரணி சம்பந்த அட்டவணையாய் மாற்றி அதன் IQ'வை கணக்கிடுறோம்.

கணிமையின் சமீபகால போக்குகளைப் கவனித்தால், செயற்கை நரம்பணுப் பிணையம் பயன்பாடுகள் மூலம் ஒரு வடிவமைக்கப்படாத மென்பொருள் எழுச்சி பார்க்கலாம். எனவே, இயற்கையான அடுத்த படி ஒரு வடிவமைக்கப்படாத வன்பொருள் எழுச்சி என்று எதிர்பார்த்தால், பிறக்கக்கூடும் ஒரு எந்திரன் (ENTHIRAN: evolved network that has intelligence realized at nanoscale) •

THE WORLD IS NOT ONLY FOR HUMANS...

- (ENTHIRAN) 2.0 

The first large-scale electronic computer was built in times of the Second World War. Named as ENIAC (electronic numerical integrator and computer), the machine was as big as a stack of 100 humans. It was originally designed to calculate firing tables for artillery, a task humans were bad at. Fortunately, the war ended and subsequent generations of computers have enhanced many aspects of human activity and helped us achieve otherwise impossible tasks such as space exploration. Modern computers have shrunk to microprocessors smaller than the size of a thumb, and thus become much more quick and energy efficient. This technological advancement has made computers capable applications like Microsoft's *Seeing AI* which can narrate the scenery captured on camera to enrich the lives of the blind. This kind of AI (artificial intelligence) is made possible due to *deep learning* algorithms on artificial neural networks (which are mathematical models of computational networks similar to the brain). There is a buzzing development of computer chips that can efficiently implement artificial neural networks, and deep learning machines have recently achieved superhuman skills in playing strategic board games and video games, and have brought new knowledge and intuition to play these games. This is done by optimizing millions of synaptic weights in an artificial neural network, by *backpropagation* of errors in decision by adjusting synaptic weights layer-by-layer; going backward from the output layer through hidden layers and finally to the input layer. However, we are still a moonshot away from achieving what is known as artificial general intelligence, which also includes empathy and wisdom to make decisions between good and evil in the real world.

The challenge is to make *neuromorphic* hardware that is as large-scale as the human brain which consists of  $10^{11}$  neurons with an average of  $10^4$  synapses per neuron. In comparison, IBM's *TrueNorth* which has been the largest neuromorphic chip over the last four years, has only  $10^6$  neurons and 256 synapses per neurons, implemented on a thumb-sized chip packing  $5.4 \times 10^9$  transistors based on 28 nm CMOS technology. Shrinking the feature sizes towards the bottom of the nanoscale, to achieve greater computational density, leads to imperfections and other unintended physical effects, thus beyond the scope of human-design. To resolve this, we recall the ways of nature!

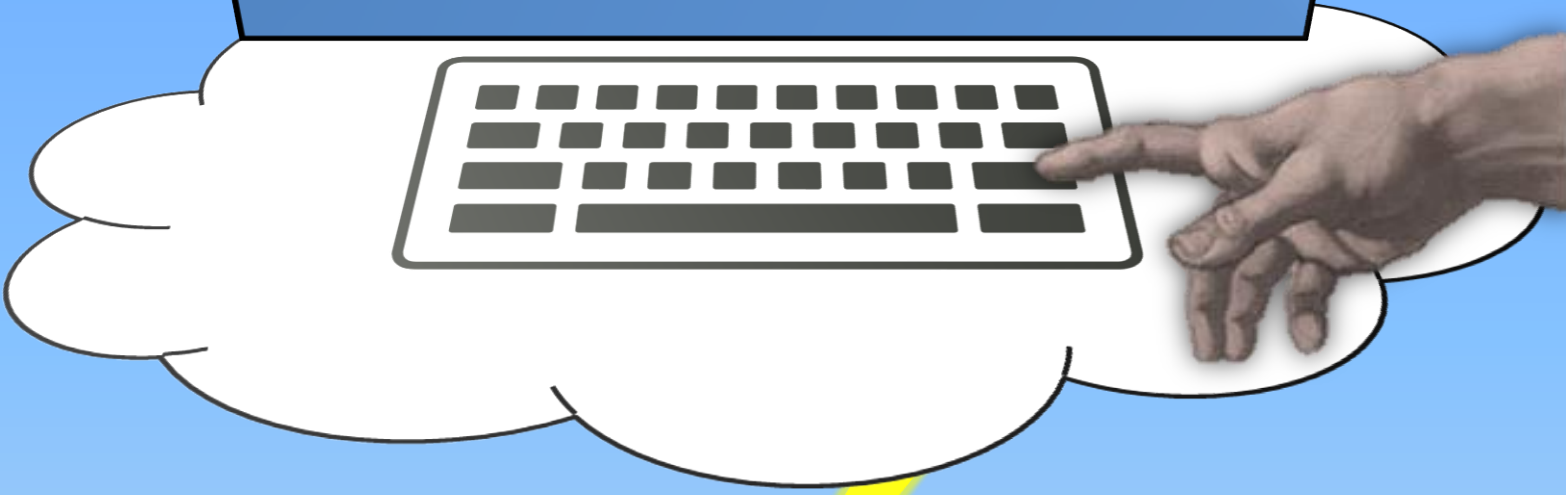
*Natural computing* systems are a product of evolution. Evolution uses whatever physical processes are exploitable. So, instead of making electronic circuits that have design rules to exclude physical processes such as capacitive crosstalk, it seems more natural to use matter in its *designless* form and evolve it for computation. Here, we show that the electronic properties of nanomaterial clusters can be evolved to realize Boolean logic gates. The cluster is designless in the sense that it is a disordered assembly of particles. Clusters of nanoparticles are tested at 300 mK and it is discovered that our system meets the essential criteria for the physical realization of neural networks: universality, compactness, robustness and evolvability. Similarly, clusters of atoms doped in silicon function at the temperature of liquid-nitrogen. Prospective nanomaterials for room-temperature functionality are given.

For functionality such as image recognition, we design a large-scale *small-world* architecture that interconnects several nanomaterial clusters by electronic switches whose configuration is to be evolved by a computer server. By modelling the nanomaterials system as a single-electron tunnelling (SET) network, we find in simulations that the system is evolvable to realize inhibition, an important neural network property. For simulations of large-scale SET networks, we develop a novel mean-field approximation that is typically 10-100 times quicker than a standard Monte Carlo method for simulating the current-voltage relations.

In our quest to realize *intelligence*, it is useful to have a mathematical definition and a practical measure. To that end, we model intelligence as the capacity to relate patterns to data. We map input-output relations of a physical system to a table of datum-pattern relations, from which we calculate an IQ metric.

Reflecting upon recent trends in computing, we witness a designless software revolution with the rising applications of artificial neural networks. So, it is only natural that a designless hardware revolution follows, and hopefully gives birth to an ENTHIRAN (evolved network that has intelligence realized at nanoscale) •

Evolving Networks  
To Have Intelligence  
Realized At Nanoscale



CELESTINE ப்ரீதம்