Omnipotent and Mortal Training of a Nanocell Model to Emulate the Functionality of a Logic Gate

Pankaj Kumar Jha[1], Vineet Sahula[2]

Department of Electrical Engineering, IIT Hyderabad
Hyderabad, India
[1]jovialjha@gmail.com

Department of Electronics and Communication Engineering, M.N.I.T Jaipur
Jaipur,India
[2]sahula@ieee.org

Abstract—**Various unavoidable constraints (viz. physical, technical and financial) curtail the possibility of achieving continuous improvement in the computing capabilities through scaling down of devices using the conventional silicon technology. Molecular electronics aims to use the bottom-up approach to build nanoscale devices from basic molecular unit and promises unforeseen levels of computing per dollar-watt-cm^2. The programmability feature of molecules is exploited to circumvent the problem of addressability. The nanocell concept is predicated on the belief that a random distribution of self-assembled molecules can be programmed to perform a specific logic function. In this paper we present a novel approach to demonstrate plausibility of the idea of "creating functionality from disorder". The experimental results vindicate the plausibility of training a nanocell to perform a logic operation. A negative differential resistance (NDR) circuit has been designed to emulate the Λ-type I-V characteristics of the molecular switches connected between any pair of nodes in the actual nanocell. A nanocell model is then constructed taking instances of this NDR circuit. As a primary exploration of the nanocell concept the omnipotent programming was considered. The results from HSPICE simulations are then fed to the genetic algorithm(GA) solver in MATLAB to provide us with the optimized configuration(or a combination of switch states) of the NDR circuits for which the nanocell model yields the functionality of one or multiple target logic devices. Finally mortal programming is also accomplished. The GA solver is used again to provide us with the voltages which ought to be applied on each of the exterior nodes (apart from the input and output nodes) of the nanocell to yield a response resembling a NAND gate.**

***Keywords*— Beyond CMOS, nanocell, molecular electronics, genetic algorithm, negative differential resistance, omnipotent, mortal, programmability.**

I. MOTIVATION: CREATING FUNCTIONALITY FROM DISORDER

The branch of moletronics aims to extend electronics beyond any contemporary addressable lithographic techniques. The International Technology roadmap for Semiconductors, released by Semiconductor Industry Association portends that the conventional Si technology would reach some fundamental limits by the year 2016, and points out to the implementation of 'beyond CMOS' devices for future technology generations [7]. It can be attributed to the numerous unavoidable obstacles curtailing the continuous downsizing of Si. These factors include financial constraints apart from the technical constraints e.g. tunnelling of electrons, fuzziness between the sharp on and off states at nanoscale, heat dissipation and removal, strong electron perturbations, problem of fabrication and addressing, etc. [2].

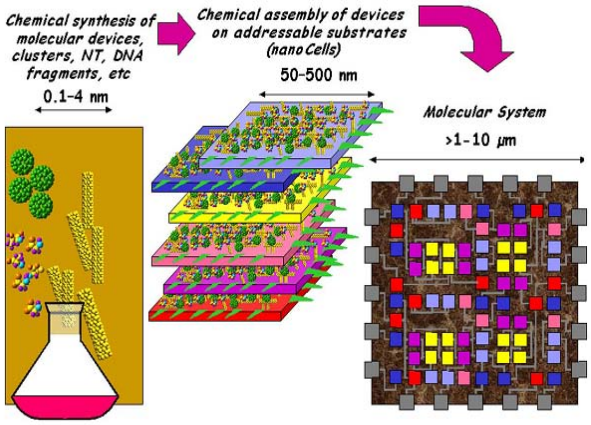


Fig. 1 Possible bottom-up approach to the fabrication of a nanocell IC on a top-down substrate [2]

Molecular electronics alleviates the above mentioned financial and physical constraints, since devices are fabricated using bottom-up approach, as shown in Fig.1. Individual or arrays of molecules perform as a device. Competing molelectronics architectures like the crossbar and quantum cellular automata require precise molecular order and building arrays of logic exact arrays of nanostructures. The programmable nanocell approach circumvents this problem. Molecular circuits need not be assembled deterministically but most of the molecules are randomly interconnected and oriented. Owing to the small size of the molecular circuit, no probes can access the individual molecules. This problem of addressability is circumvented by exploiting the programmability feature of the molecules. The nanocell consists of self-assembled molecules attached to the nanoclusters, and programming is achieved by applying a series of voltage signals to the exterior microscopic pads.

II. THE NANOCELL PARADIGM

A nanocell is actually a smallest addressable chip constructed using the available addressable technology (Fig.2a). The nanocell consists of molecules which are based upon nitro-containing oligos (Fig.2b) and exhibiting reprogrammable (can be turned on and off) NDR characteristics. NDR characteristics are necessary to provide the negating functionality e.g. NAND and NOT gate. These molecules are chemically self-assembled and randomly interconnected to each other through nanoclusters using gold atoms with an oxide layer acting as an insulating base. The molecules can be considered as two terminal devices endowed with special "allegator clips" such as thiols enabling them to bond directly to the metal clusters and not to the substrate or other molecules. The spacing between the nanoparticle is controlled by the self-assembled monolayer of short alkanethiols coating each nanoparticle, preventing them from coalescing into a multi-particle array. Adequate numbers of lithographically defined access pads are provided at the edges of the nanocell. A typical nanocell would contain 250-1000 nanoparticles and 750-10,000 molecular switches approximately [3].

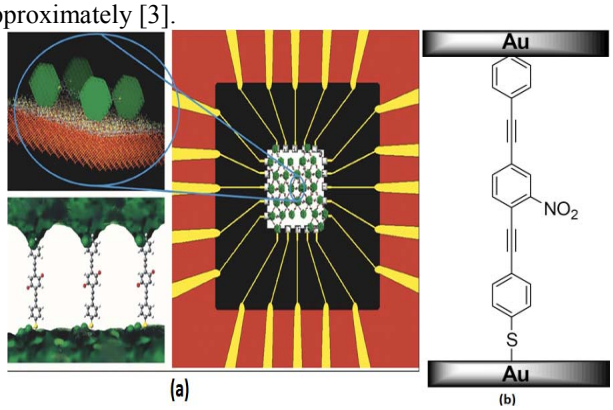


Fig. 2 (a) A moletronics system which consists of self-assembled molecules attached to clusters [1]. (b) The chemical structure of nitroaniline which is used as a switch and is connected between any pair of nodes in the nanocell[4].

The physical topology (which is mostly disordered) is assumed to be static in the sense that the physical location of each individual molecule switch is fixed, the existing nodes cannot be deleted and new nodes and edges cannot be added. The only amendable parameter is the switch states of the molecular switches (conducting on or non-conducting off), as is set by the voltages applied on the exterior access leads (mortal programming) or by the experimenter himself (omnipotent programming). Logic is emulated through the nanocell post fabrication, by exploring the *on* and *off* states of the molecular switches until the nanocell behaves as the target logic device. The training of the nanocell may be accomplished by using either of the assumptions, omnipotence and mortal switching.

The task of training a nanocell, omnipotently or mortally, can be formulated as an optimization problem. Here the objective would be to make a specific nanocell emulate the functionality of a target logic gate. The optimization process would require as an input the nanocell to be trained, the target logic in the form of a truth table and the I-V characterstic of the *on* state and *off* state of the molecular switch. Search space for omnipotent training would include all possible combinations of *on* and *off* molecules and all possible assignments of input and output pins. Similarly, search space for mortal training would include all possible sets of values corresponding to the voltages that ought to be applied on the exterior access leads so that the nanocell yields the desired response and all possible assignments of input and output pins. Reference [4] provides an intuitive proof in support of the probabilistic feasibility of extracting logic from the random arrangement of nanoparticles, and issues pertaining to the reliability of these nanocells.

Before attempting to explore the nanocell concept using mortal training, it is logical to explore the possibility of training the nanocell omnipotently. If the training with the assumption of omnipotence is unsuccessful, it implies that mortal training is impractical. As mentioned above, a typical nanocell contains 750-10,000 molecular switches. This implies that the search space for the omnipotent training is minimally 2^{750}. It is impractical to explore all the 2^{750}combinations of states. GA is used to circumvent this problem of a huge search space. The GA proves to be efficacious for mortal training of the nanocell too, although the number of exterior nodes is very small comparatively. Here we report results showing successful omnipotent and mortal training.

III. MODELING A NANOCELL AND ITS GA BASED TRAINING

The stepwise summary of the procedure used to model and train a nanocell is shown in Fig.3. The details are discussed later in the related subsections which follow.

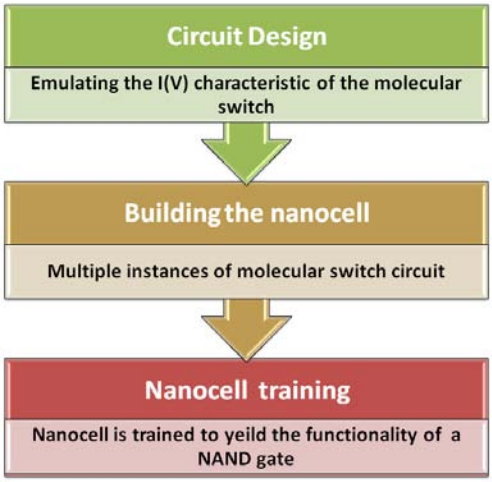


Fig. 3 Flowchart indicating modeling of the nanocell and its training using GA

A. Design of a circuit emulating the Λ-type I–V characteristic of a molecular switch

To start building the nanocell model, a NDR circuit with Λ-type I-V curve is first designed. The steps followed for the designing of such a circuit are illustrated in the Fig.4. The basic structure of the circuit is adopted from Kwang et al.[5].

In this report we present design of a circuit yielding a Λ-type I–V characteristic, using the combination of the standard Si-based n-channel metal-oxide-semiconductor field-effect-transistor (NMOS) and SiGe-based hetero-junction bipolar transistor (HBT). The circuit is simulated using HSPICE. The desired I-V curve is obtained after a few trials exploring various circuit parameters in the circuit.

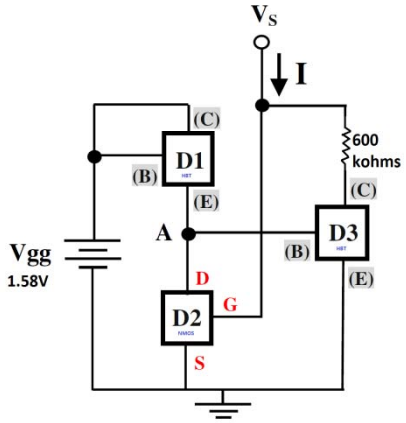


Fig. 4 The NDR circuit

The operation states of the devices Q1, M1, and Q2 under different I–V regions are summarized in Table 1.

TABLE I
OPERATION CONDITIONS FOR THE CIRCUIT WITH Λ-TYPE I–V CURVE

	Q1(HBT)	M2(MOS)	Q2(HBT)
Positive Differential region	Diode → ON	OFF → SATURATION	OFF → SATURATION
Negative Differential Region	Diode → ON	SATURATION → TRIODE	SATURATION → TRIODE
Constant High Resistance Region	Diode → ON	TRIODE	OFF

The target I-V curve proposed in Tour et al.[3], and the experimentally obtained I-V curve of the above NDR circuit are shown in Fig.5 and Fig. 6 respectively.

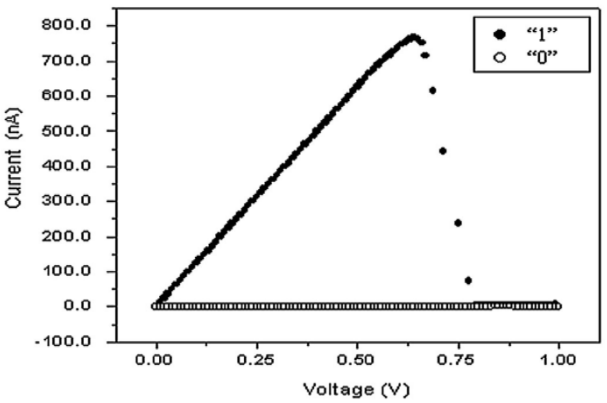


Fig. 5 The target I-V characteristic [3]

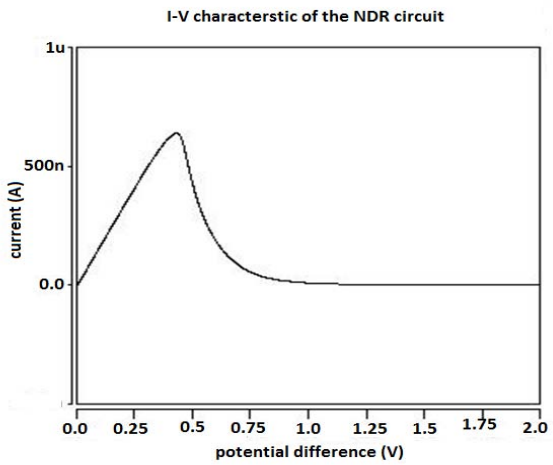


Fig. 6 Experimentally obtained Λ-type I-V curve after simulation of the NDR circuit

The above figures clearly show that the I-V curve obtained after simulation of the NDR circuit closely matches with the target I-V curve as proposed in Tour[3].

B. Building the nanocell using the NDR circuit

Following the design of the circuit with I-V characteristic resembling that of the molecular switches, the next step is to build a nanocell using this circuit. A conceptual model of a nanocell would be a circuit consisting of components with I-V curve similar to that of the real molecular switches. The NDR circuit, discussed in the previous subsection, is used as a subcircuit. The component connected between any two nodes is an instance of this subcircuit. A nanocell is then constructed taking 242 instances of the subcircuit. The netlist of the nanocell hence obtained is simulated using HSPICE for the voltage in-current out model of the nanocell. The simulation results of the netlist of the complete nanocell ensure that the nanocell is ready for training with GA. The omnipotent training procedure is discussed in the following subsection.

C. Training of the nanocell with GA

To start with, random strings of 242 bits are generated, representing the 242 instances of the NDR circuit. The strings constitute the `population'. Each bit-string constitutes a 'chromosome' or an 'individual', representing a set of switch states (on or off) of the molecular switches connected between any two nodes in the nanocell model. A '1' value in the string at i^{th} position indicates presence of i^{th} NDR, whereas a '0' in the j^{th} position indicates absence of j^{th} NDR. The nanocell is then simulated using HSPICE. The values of the output currents are then used to calculate the fitness scores of each of the individuals. Fitness scores are calculated taking I_{OH} = 600nA, I_{OL} = 300nA, V_{OH} = 2v and V_{OL} = 0.5V. The methodology of evaluating the fitness scores is based on the calculation of the magnitude of deviation from the expected/desired response. When the desired logic level is '0', the case for zero deviation would be when the measured

current is less than or equal to the fixed I_{OL}. If the measured current turns out to be greater than I_{OL}, the difference between the two currents would give the error. For the case where the expected logic level is '1', the case of zero deviation would be when the measured current is greater or equal to I_{OH}. If this criterion is violated, the difference between the two currents would give the error. Similar calculation is done for each row of the truth table corresponding to the target logic. Net fitness score is the summation of the errors calculated from each row of the truth table. After the fitness of a generation is evaluated, the GA solver applies processes like selection, crossover, mutation, elitism's, etc. to create subsequent generations. The GA solver provides us with a set of switch states, or a configuration of the nanocell, for which the behaviour of the nanocell matches most closely to the target logic gate. Note that a fitness of zero would indicate that the individual successfully functions as the target logic device.

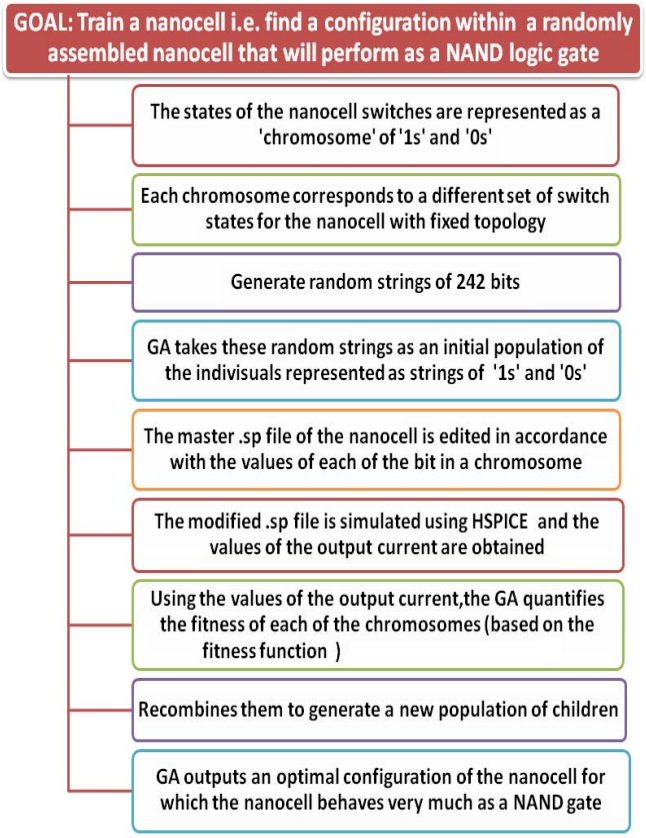


Fig. 4 Procedure to train a nanocell using GA solver

The tuning of GA solver is highly desired for faster convergence. The *optimset* command in MATLAB creates a structure called options that contains the options, or parameters, for the genetic algorithm, and is used to set values for parameters whose values are required to be different from the default values. An important step in training the nanocell is applying the aforesaid procedure for different combinations of the triplet consisting of the two input nodes and one output node.

The GA solver provides us with the switch states for which the nanocell behaves like a NAND gate. The nanocell with the obtained switch states is simulated to verify the results. Fig. 8 shows the waveforms of the inputs applied and the corresponding output currents obtained for training the nanocell. A voltage of 2V is applied whenever a '1' appears in the truth table, a voltage of 0.5V is applied wherever a '0' appears in the truth table, output is considered logic '1' if output current ≥600nA and logic '0' if ≤300nA .

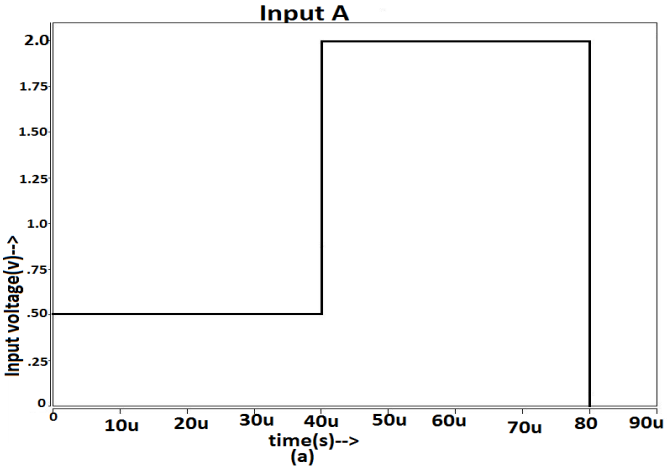


Fig. 8 (a) Input voltage A

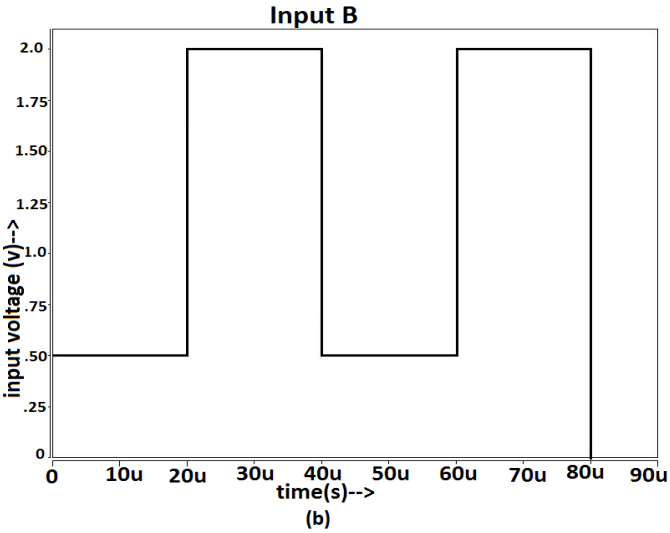


Fig.8(b) Input B

For all the simualtions discussed in this paper, the voltage-in current-out model of the nanocell has been used. The waveforms of the two input voltages (here Input A and Input B) and the output current obtained after the simulation of the nanocell model with the configuration suggested by the GA solver are shown in Fig.8. The corresponding plots of best function value versus generation and scores of the individuals

at each generation are also presented in Fig.9 and Fig.10 respectively.

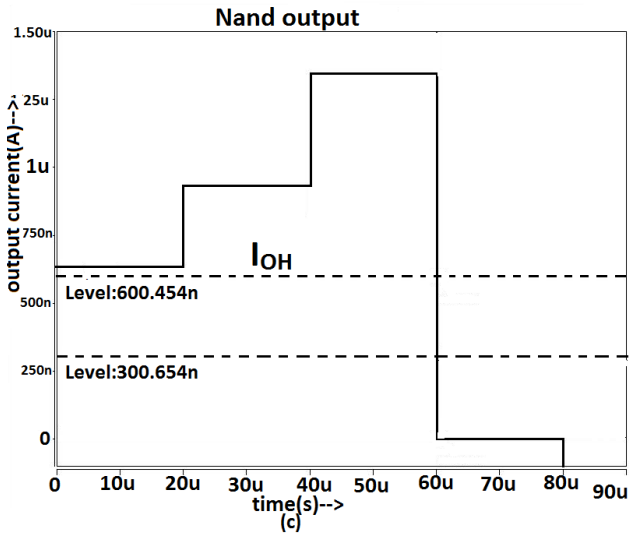


Fig. 8(c) Output waveform of the nanocell trained as a NAND gate

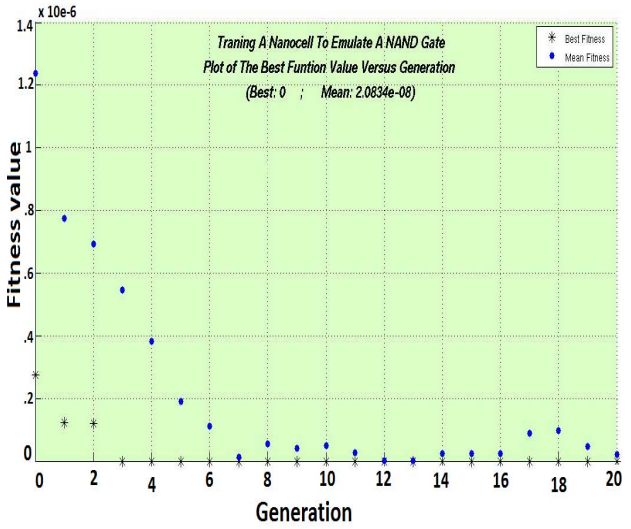


Fig. 9 Plot of the best function value vs. generation

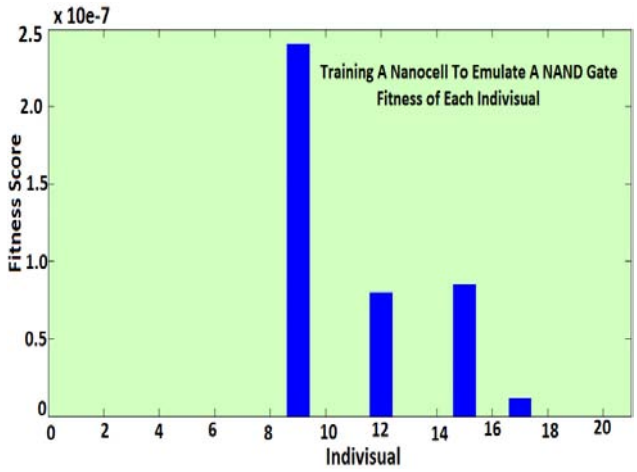


Fig. 10 Plot of the fitness scores vs. individual

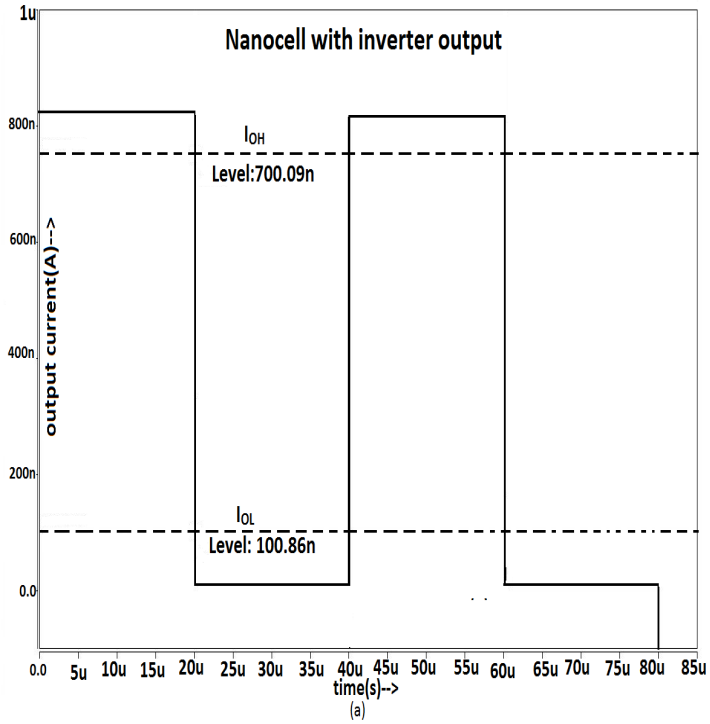
D. *Defect- and fault-tolerance*

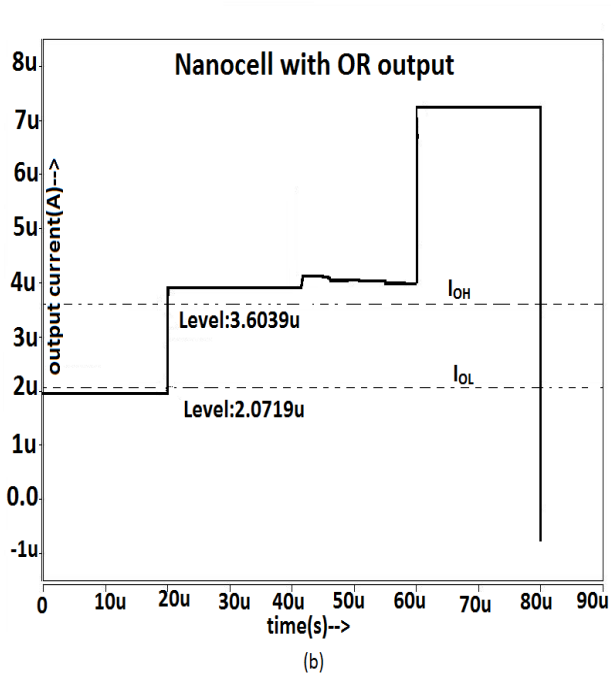
Fig. 10 shows that the fitness score is zero for a number 'individuals'. This indicates that a nanocell trained as a NAND is defect tolerant because the performance of the logic gate does not depend on a single set of switch states. Furthermore, as a check of defect tolerance, the nanocell trained as a NAND i.e. nanocell with the configuration suggested by the GA solver , is tested for variation in the I(V) curve of the NDR circuit emulating the molecule used in the nanocell. After training the nanocell omnipotently, exploration of the nanocell concept using mortal training was also attempted. The nanocell was trained with a peak current at 0.431V. The nanocell maintains its NAND functionality for peak voltages of 0.395V to 0.51052V, a considerably wide range of molecular peak voltages.

E. *Multiple logic gates from a single nanocell*

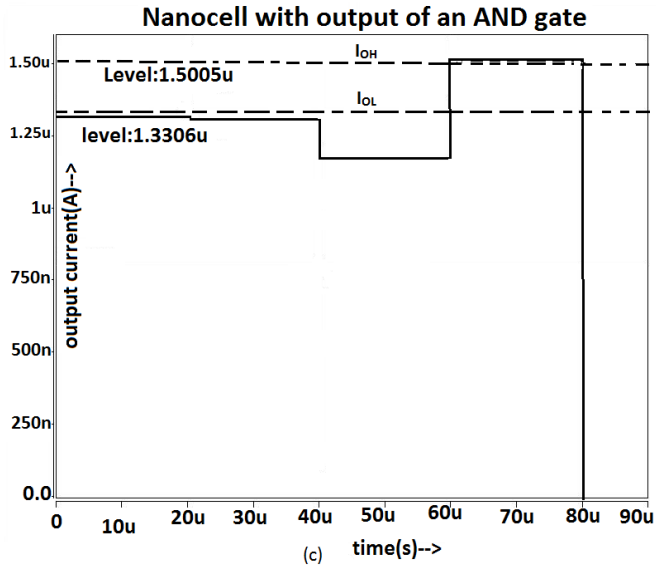
An interesting extension of the above experiments was to try and explore the possibility of extracting the behaviour of multiple logic gates from a single nanocell. Here we chose the functionally complete set of logic gates, i.e. AND, OR and NOT, and tried to train the nanocell to yeild the functionality of all the three gates simultaneously. Again we will have to try all possible combinations of nodes which are declared as input nodes and those declared as output nodes. For the OR and AND gate the two inputs are same as that shown in Fig.8a and Fig.8b, and the input for the inverter is that shown in Fig. 8b.

The results of this experiment point towards the possibilities of reconfigurability of the nanocell.





(b)



(c)

Fig. 11 Output waveforms: (a) NOT gate (b) OR gate (c) AND gate

F. Mortal switching

After training the nanocell omnipotently, exploration of the nanocell concept using mortal training was also attempted. The GA solver is used again to provide us with a set values corresponding to the voltages that ought to be applied on the exterior pads so that the nanocell yields the response of a NAND gate. This goal was achieved successfully, proving the possibility of mortal training of the nanocell.

IV. Conclusions and Future Trends

The prime motive of this paper was to demonstrate the proof-of-principle possibilities for a nanocell using omnipotence. A NDR circuit exhibiting an I(V) curve similar to that of the actual molecular switches was designed. The structure of the nanocell was built using these NDR circuits. This nanocell was then successfully trained omnipotently using GA solver to yield the functionality of a NAND gate. The simulations indicated that it will probably not be difficult to mortally train a nanocell. Also, the nanocell was tested for defect tolerance. The results show that the nanocell trained as a NAND is defect tolerant. The same nanocell is then trained to yield the functionality of three independent logic gates viz. NOT, AND and OR. It further vindicates the possibility of extracting multiple gates from a single nanocell. Finally the idea of mortally training a nanocell has been explored. The results strongly support the plausibility of training a nanocell, where programming controls are limited to only the exterior of the nanocell, without having any apriori knowledge of its internal topology. Future challenges in this field would include finding methods to train a given nanocell in minimum time. Studies should aim to answer questions like what is the maximum complexity of a logic function that can be extracted from a nanocell, or , what is the minimum size of a nanocell which can yield the functionality of a logic gate of a given complexity.

References

[1] Jorge M. Seminario, Luis E. Cordova, Pedro A. Deros, “An Ab Initio Approach to the Calculation of Current–Voltage Characteristics of Programmable Molecular Devices,” Proceedings of the IEEE, Vol. 91, No. 11, November 2003.

[2] Yuefei Ma, and Vandana Tarigopula, Jorge M. Seminario, “ The Nanocell: A Chemically Assembled Molecular Electronic Circuit,” IEEE Sensors Journal, Vol. 6, No. 6, December 2006.

[3] J.M. Tour, W.L. Van Zandt, C.P. Husband, S.M. Husband, L.S. Wilson, P.D. Franzon, D.P. Nackashi, “ Nanocell logic gates for molecular computing,” IEEE Transactions on Nanotechnology.1(2) , pp. 100 – 109, 2002.

[4] Summer Michele Husband, “ Programming the nanocell, a random array of molecules,” PhD thesis-935569, isbn = {0-493-61528-8},2002.

[5] Kwang-Jow Gan , Cher-Shiung Tsai, Dong-Shong Liang, “ Design and characterization of the negative differential resistance circuits using the CMOS and BiCMOS process,” Analog Integrated Circuits and Signal Processing, pp. 63-68, 2010.

[6] Marcea R.Stan, Seth Copen Goldstein, Matthew M. Ziegler, “ Molecular Electronics: From Devices and Interconnects to Circuits and Architecture,” Proceedings of IEEE, Vol. 91, No.11, November 2003.

[7] International Technology roadmap for Semiconductors: 2009. 2009, Semiconductor Industry Association.

[8] MATLAB R2008b Product Help.