

Evolution of a designless nanoparticle network into reconfigurable Boolean logic

S. K. Bose^{1†‡}, C. P. Lawrence^{1,2‡}, Z. Liu¹, K. S. Makarenko¹, R. M. J. van Damme³, H. J. Broersma² and W. G. van der Wiel^{1*}

Natural computers exploit the emergent properties and massive parallelism of interconnected networks of locally active components^{1–3}. Evolution has resulted in systems that compute quickly and that use energy efficiently, utilizing whatever physical properties are exploitable⁴. Man-made computers, on the other hand, are based on circuits of functional units that follow given design rules^{5,6}. Hence, potentially exploitable physical processes, such as capacitive crosstalk, to solve a problem are left out^{7,8}. Until now, designless nanoscale networks of inanimate matter that exhibit robust computational functionality had not been realized. Here we artificially evolve the electrical properties of a disordered nanomaterials system (by optimizing the values of control voltages using a genetic algorithm) to perform computational tasks reconfigurably. We exploit the rich behaviour that emerges from interconnected metal nanoparticles, which act as strongly nonlinear single-electron transistors^{9,10}, and find that this nanoscale architecture can be configured *in situ* into any Boolean logic gate. This universal, reconfigurable gate would require about ten transistors in a conventional circuit. Our system meets the criteria for the physical realization of (cellular) neural networks¹¹: universality (arbitrary Boolean functions), compactness, robustness and evolvability, which implies scalability to perform more advanced tasks^{12,13}. Our evolutionary approach works around device-to-device variations and the accompanying uncertainties in performance. Moreover, it bears a great potential for more energy-efficient computation, and for solving problems that are very hard to tackle in conventional architectures^{14–16}.

As many physical and chemical processes can be thought of as a computation, it seems natural to use matter itself for computation¹⁷. Miller and Downing proposed to use the artificial evolution of complex physical systems for this purpose, and introduced the term ‘evolution in materio’¹⁸. Earlier attempts to evolve (microscale) material systems into computational units encountered serious difficulties concerning reproducibility, recoverability, robustness and stability¹⁹. Although theoretical proposals were made for the evolution of nanoscale matter, for example, logic realized by nanoparticles (NPs) interconnected by molecular switches²⁰, experimental realization of ‘evolution in nanomaterio’ has remained elusive so far.

Here we demonstrate experimentally that a disordered NP network can be configured *in situ* into any of the two-input Boolean logic gates by tuning six static control voltages. This is similar to the approach of Tour *et al.*²⁰, wherein they show, through simulations, that a ‘nanocell’ may be programmed by

tuning voltages at its periphery. However, they used a network of tunable molecular switches with the NPs serving as interconnects. In our work, we exploit the rich behaviour that emerges from 10–100 arbitrarily interconnected NPs, which act as strongly nonlinear single-electron transistors (SETs)⁹.

That our system is truly designless and reconfigurable makes our approach fundamentally different from the designed circuits of SETs^{10,21,22} or NPs²³ that were used to realize logic units previously, and also from massively parallel (but still design constrained, see the Supplementary Information) architectures such as HP’s Teramac^{24,25} or IBM’s TrueNorth²⁶.

Figure 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/SiO₂ 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 SET⁹ (Fig. 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. Two electrodes are used to apply time-varying voltage signals V_{IN1} and V_{IN2} and the remaining six electrodes are used to apply static control voltages $V_1–V_5$ and to measure the resulting current I_{OUT} . In addition, a static voltage V_6 is applied to the back gate.

We find that electron transport below ~ 5 K is dominated by Coulomb blockade, and strongly depends on the input and output electrodes used, as well as on the static voltages $V_1–V_6$ applied to the remaining electrodes (Supplementary Fig. 1). Based on the strongly nonlinear (switching) behaviour of the SETs and their mutual interactions, we set ourselves the goal to find logic gates, as schematically depicted for an AND gate in Fig. 1c. With six control voltages we have a huge six-dimensional search space, which renders a brute-force full search very time-consuming (order of days). 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 annealing²⁷. However, as our search landscape is strongly non-monotonic (Supplementary Fig. 3c), we

¹NanoElectronics Group, MESA+ Institute for Nanotechnology, University of Twente, PO Box 217, Enschede 7500 AE, The Netherlands. ²Programmable NanoSystems and Formal Methods and Tools, MESA+ Institute for Nanotechnology and CTIT Institute for ICT research, University of Twente, PO Box 217, Enschede 7500 AE, The Netherlands. ³Multiscale Modeling and Simulation, MESA+ Institute for Nanotechnology, University of Twente, PO Box 217, Enschede 7500 AE, The Netherlands. [†]Present address: Department of Physics and Astronomy, University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand. [‡]These authors contributed equally to this work. *e-mail: w.g.vanderwiel@utwente.nl

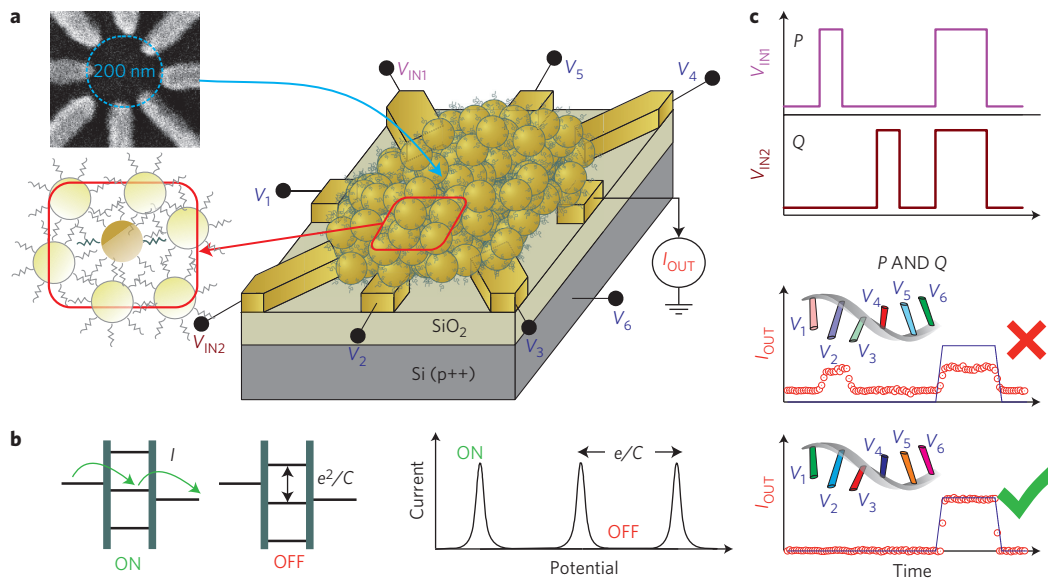


Figure 1 | Schematic of the device layout and working principle. **a**, Disordered network of ~ 20 nm Au NPs separated by ~ 1 nm 1-octanethiols in between eight Ti/Au electrodes, shown in the scanning electron micrograph (top inset). Two time-varying input-voltage signals V_{IN1} and V_{IN2} and six static control voltages V_1 – V_6 give rise to an output current I_{OUT} . **b**, The Au 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 $V_{IN(2)}$ of the form P (Q) generate an output current I_{OUT} , which represents an AND ($P \cdot Q$) in the cases of a ‘bad’ (red cross) or a ‘good’ (green tick mark) set of control voltages V_1 – V_6 , schematically represented by strands of genetic information shown in the insets.

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 utilized²⁸.

We draw an analogy with how genes determine the functionality of an organism, and consider each control voltage as a gene, and hence the complete set of control voltages (V_1 – V_6) as a genome (see insets to Fig. 1c). The full genome determines what the output signal I_{OUT} looks like for given input signals V_{IN1} and V_{IN2} . How well the input–output signal combinations represent the desired functionality (for example, an AND gate in Fig. 1c) is reflected by the fitness of a certain genome, and is calculated using a fitness function F (see Methods). To preserve the best-performing (that is, fittest) genomes and yet search for better-performing candidates, a composite cloning–breeding approach was used, as described in detail in the Methods. Starting with a generation of randomly generated genomes (with usually low fitness), desirable traits are passed selectively on to subsequent generations and ‘elite’ genomes converge towards higher fitness as the GA progresses. The GA halts after a satisfactory fitness level is reached.

Using the artificial evolution procedure described above, we succeeded to realize fully reconfigurable, robust Boolean logic in disordered NP networks at our base temperature of ~ 0.3 K. These results comprise the first experimental demonstration of exploiting disordered matter at the nanoscale for computational functionality. Similar results were obtained in three independent NP networks on separate chips, but in the following discussion we focus on the NP network shown in Fig. 2a. Based on the measured resistance, and assuming regular connectivity, we estimate that typically a few tens of NPs participate in electron transport between the electrodes (see Supplementary Fig. 6). The input signals V_{IN1} and V_{IN2} of forms P and Q (Fig. 2b) are applied to two electrodes. The remaining electrodes are used to apply five control voltages and reading out I_{OUT} . The sixth control voltage is applied to the doped substrate. Figure 2c shows the I_{OUT} versus time (t) plots that correspond to the inverter (NOT) gates \bar{P} and \bar{Q} . The I_{OUT} versus time (t) plots for the six most-commonly used two-input

Boolean logic gates are given in Fig. 2d. Importantly, all the logic gates were obtained in the same network with the same input signals V_{IN1} and V_{IN2} , but with different evolved genomes.

For each logic gate, the GA almost always converged to a viable genome (defined by a fitness much larger than one) in less than 200 generations, with the genome having a few preferential gene values (see Supplementary Fig. 3). Currently, this process typically takes about one hour because we use rather slow input signals. However, by optimization of the set-up this can easily be increased by a factor of 100–1,000. The back-gate voltage (gene 6) was 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 network (see Supplementary Fig. 7). This demonstrates that the same NP network with the same input signals can be reconfigured *in situ* to represent a completely different functionality. We emphasize that a conventional digital computer is only used to find the solutions with the GA, after which we only need to apply the obtained control voltages to configure the NP network into the desired logic gate. For the exclusive gates (XOR and XNOR), spike-like features are observed at the rising and falling edges of the (1,1) 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 about ten, whereas for all other gates it is ~ 20 . Interestingly, these exclusive gates are inherently more difficult to evolve, in agreement with the theoretical framework on evolvability developed by Valiant²⁹.

To see whether artificial evolution of our system has a clear advantage compared with a brute-force search for finding solutions, we recorded the output response of 19,000 randomly generated genomes, which took over seven hours (see Fig. 2e). For all 16 possible two-input-one-output truth tables (except for the trivial null (0) and identity (1) outputs, which were not searched for) we found that at least 0.1% of the randomly searched genomes are viable solutions. As shown in the inset of Fig. 2e, the GA-searched gate solutions (taking about one hour) generally have an order-of-magnitude higher abundance and about a three-times larger fitness than their randomly obtained counterparts (found within seven hours),

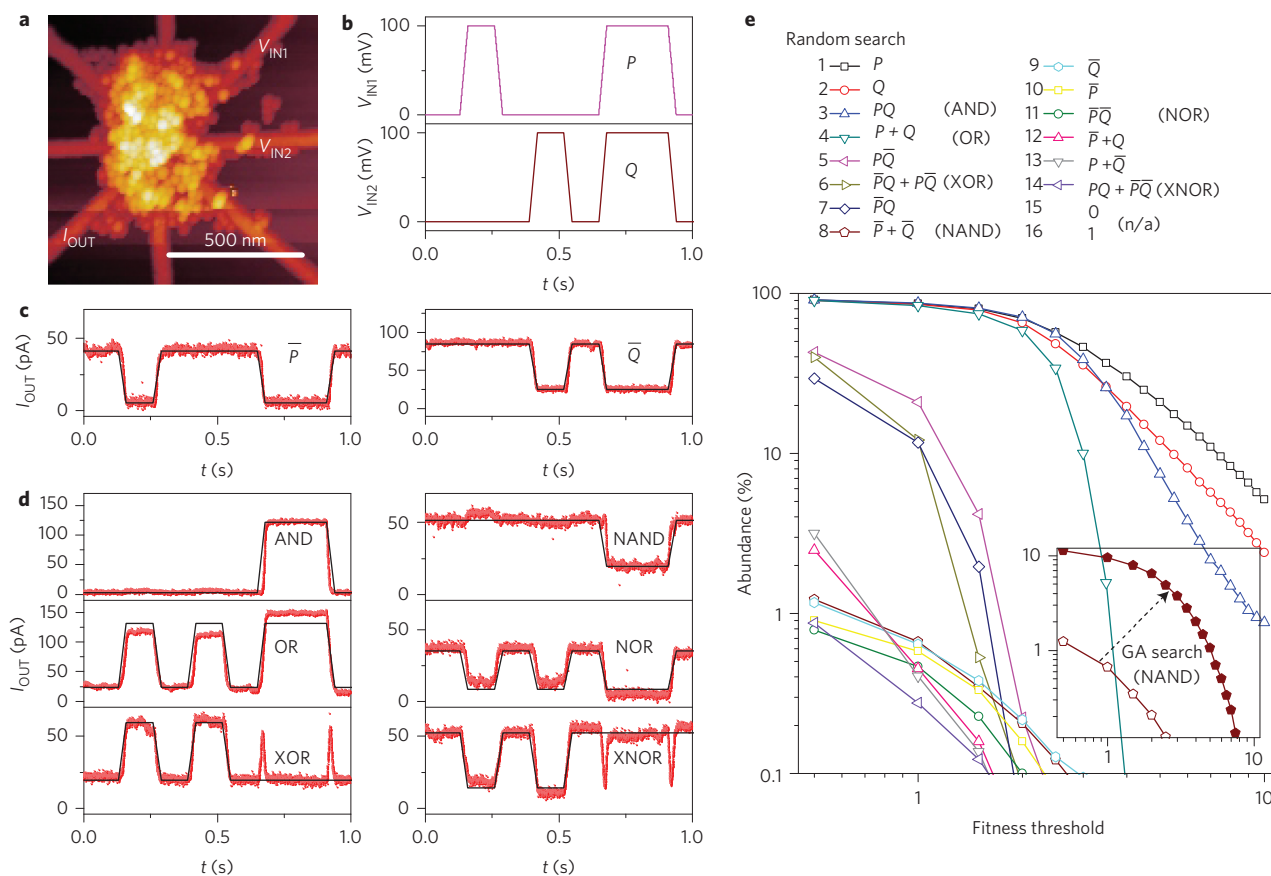


Figure 2 | Reconfigurable Boolean logic gates. **a**, Atomic force micrograph of a typical Au NP network assembled between eight electrodes. Electrodes marked $V_{IN(2)}$ and I_{OUT} are used for the signal input and output, with the remaining electrodes and the back gate for the control voltages. **b**, Time-varying input voltages $V_{IN(2)}$ of the forms P and Q . **c,d**, Inverter (NOT) gates \bar{P} and \bar{Q} (**c**) and a complete set of basic two-input Boolean logic gates (**d**), obtained by artificial evolution of the NP network at 0.3 K. Red symbols are experimental data, and solid black curves are the expected output signals (matched to the amplitude of the experimental data). **e**, Output response of 19,000 randomly generated genomes showing abundance with gate performance greater than the fitness threshold. All possible two-input-one-output truth tables searched for are found. The GA search outperforms the random search with more and better solutions. For example, for the NAND gate shown in the inset, both the abundance and fitness of the genomes is higher for the GA search (closed symbols) than for the random search (open symbols). n/a, not applicable.

which underlines the power of the GA approach chosen in our study. The relatively high abundance of fit genomes (apparent from the brute-force search) facilitates the GA convergence.

To study the fault tolerance and temporal stability of the evolved gates, we investigated the sensitivity of the output response to input- and control-voltage perturbations, thermal cycling and gate interchangeability. Interestingly, the gates function well beyond the input-voltage range they were evolved for. For example, Fig. 3a shows that a gate evolved to be an inverter for $0\text{ mV} < V_{IN1} < 100\text{ mV}$ exhibits negative differential resistance (NDR) within a considerably larger range, $-50\text{ mV} < V_{IN1} < 100\text{ mV}$. As this gate functionality is completely genome dependent, we estimated the robustness of the fittest genome (G_a) against perturbations along various gene directions, that is, control-voltage perturbations. As described in detail in Supplementary Section 2, the gate performance decays as sampled away from G_a . Interestingly, in the case of exclusive gates, this gene-dependent decay rate, away from G_a , is very high for the sixth gene, which controls the back-gate potential.

The evolved gate functionality is independent of the input waveform history, as illustrated in Fig. 3b, in which a longer input sequence is used for the AND gate from Fig. 2d. To illustrate the temperature and time stability of the gate solutions, and the robustness under switching between different gates, the NOT and AND gates were applied alternately, and the output was recorded

during warming up to 15 K (red curve) and cooling down (blue curve). As shown in Fig. 3c, the fitness decays to zero around 5 K and the output signal becomes nearly featureless (see the inset of Fig. 3c). At higher temperatures, the fitness becomes negative because the device becomes a network of linear resistors, and hence acts like a simple voltage divider (which is non-inverting). It is evident that the gate behaviour is completely recoverable for thermal cycling in this range, and that the fitness is obtained on cooling back to base temperatures. More data are given in Supplementary Fig. 9. Moreover, for temperatures kept below 15 K, the gates are robust over several days (100 hours), as shown in Fig. 3d. 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. In contrast, the Boolean logic functionality reported for microscale liquid-crystal systems was always stochastic and non-recoverable¹⁹.

Our system meets the criteria for the physical realization of a (cellular) neural network (see the Supplementary Information). With a more-sophisticated electrode layout^{26,30}, and using the same evolutionary approach, it should therefore be capable of many more computational tasks. Further experiments should quantify the scalability of our approach, and we already see some promising two-input-two-output functionality, as shown by the half-bit adder operation in Fig. 4. The offset levels could, in principle, be

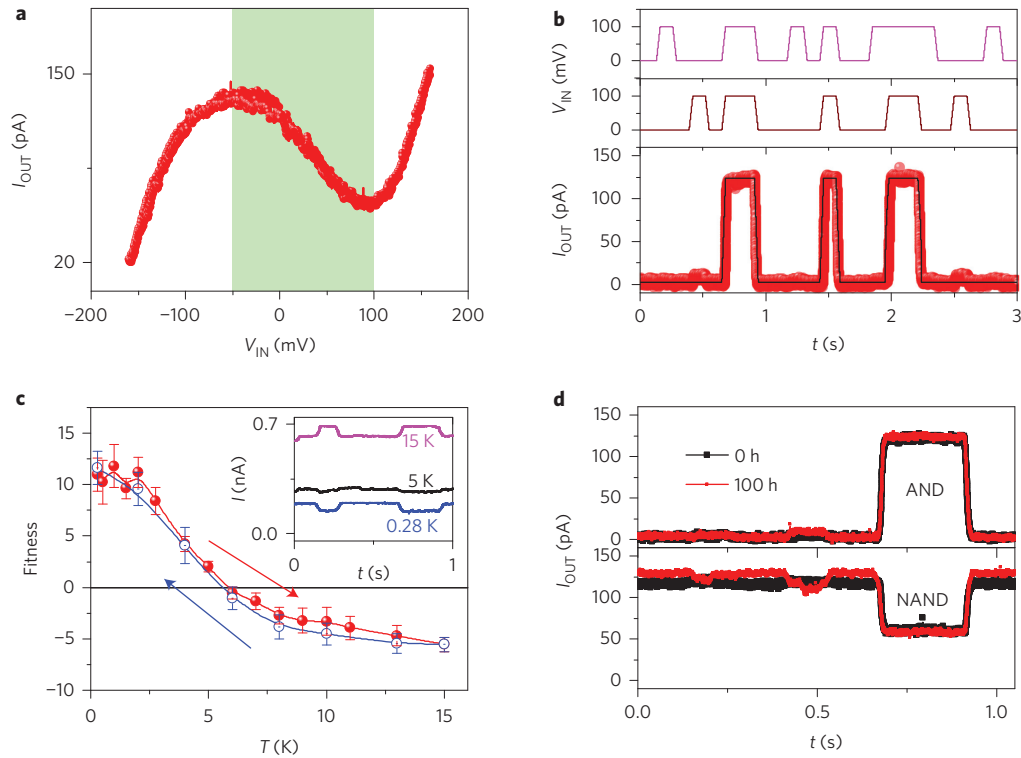


Figure 3 | Stability and robustness of logic gates. **a**, $I_{\text{OUT}}-V_{\text{IN}}$ characteristic for an inverter (\bar{P}), GA optimized for $0 < V_{\text{IN}} < 100$ mV, showing the NDR for the $-50 < V_{\text{IN}} < 100$ mV range (shaded region). **b**, For the AND gate solution of Fig. 2d, an arbitrary sequence of input signals P and Q , which correctly outputs $P \cdot Q$, which implies combinatorial independence. The red symbols are experimental data, and the solid black curves are the expected output signals (matched to the amplitude of the experimental data). **c**, Temperature fidelity tests for \bar{P} (forward and reverse sweeps) confirm the stability of the gate solutions for $T < 5$ K. The inverter and AND gate solutions were checked interchangeably, as described in Supplementary Fig. 9. The error bars represent the uncertainty in fitness for 30 measurements at each temperature. As shown in the inset, the inverter transforms into a follower (P) at higher temperatures, as the NP network then behaves as a linear voltage divider. **d**, Temporal stability of AND and NAND gate solutions after 100 hours of continuous thermal cycles and GA searches.

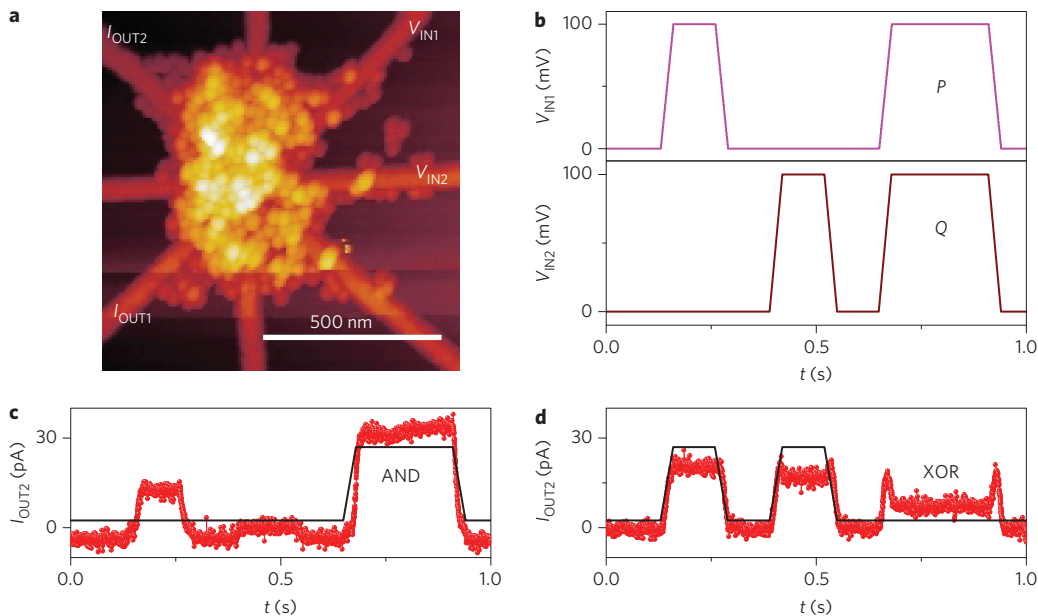


Figure 4 | Addition functionality using a two-input-two-output gate. **a**, Atomic force micrograph of an Au NP network, with electrodes marked $V_{\text{IN}(2)}$ and $I_{\text{OUT}(2)}$ used for the signal inputs and outputs, respectively. **b**, Time-varying input voltages $V_{\text{IN}(2)}$ of the forms P and Q . **c,d**, The resulting carry (**c**) and sum (**d**) from adding the input bits. Red symbols are experimental data and solid black curves are the expected output signals (matched to the amplitude of the experimental data).

improved with more control electrodes. An especially interesting avenue to explore is the suitability of this system for advanced functionality that is hard (or expensive) to realize in a conventional architecture, such as pattern recognition by mimicking brain-like systems, or simulations of complex physical systems. Our evolutionary approach works around device-to-device variations at the nanoscale and the accompanying uncertainties in performance, which is increasingly becoming a bottleneck for the miniaturization of conventional electronic circuits. The results, therefore, also need to be seen in the light of these exciting possibilities. Moreover, by choosing a smaller NP diameter, and scaling down the electrode geometry accordingly, our network may not only further reduce in area, but room-temperature operation might also appear possible.

Methods

Methods and any associated references are available in the [online version of the paper](#).

Received 9 February 2015; accepted 13 August 2015;
published online 21 September 2015

References

- Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl Acad. Sci. USA* **79**, 2554–2558 (1982).
- Watts, D. J. & Strogatz, S. H. Collective dynamics of ‘small-world’ networks. *Nature* **393**, 440–442 (1998).
- Wiesenfeld, K. & Moss, F. Stochastic resonance and the benefits of noise: from ice ages to crayfish and squids. *Nature* **373**, 33–36 (1995).
- Toffoli, T. Nothing makes sense in computing except in the light of evolution. *Int. J. Unconv. Comput.* **1**, 1–29 (2005).
- Goldstine, H. H. & Von Neumann, J. in *John von Neumann Collected Works Vol. 5* (ed. Taub, A. H.) 1–32 (Macmillan, 1963).
- Turing, A. M. On computable numbers, with an application to the Entscheidungsproblem. *J. Math.* **58**, 345–363 (1936).
- Conrad, M. *The Price of Programmability* (Springer, 1995).
- Lloyd, S. Ultimate physical limits to computation. *Nature* **406**, 1047–1054 (2000).
- Likharev, K. K. Single-electron devices and their applications. *Proc. IEEE* **87**, 606–632 (1999).
- Wasshuber, C. *Computational Single-Electronics* (Springer, 2001).
- Dogaru, R. *Universality and Emergent Computation in Cellular Neural Networks* (World Scientific Series on Nonlinear Science Series A 43, World Scientific, 2003).
- Bandyopadhyay, S. & Roychowdhury, V. Computational paradigms in nanoelectronics: quantum coupled single electron logic and neuromorphic networks. *Jpn. J. Appl. Phys.* **35**, 3350–3362 (1996).
- Asai, T. & Oya, T. in *Artificial Life Models in Hardware* (eds Adamatzky, A. & Komosinski, M.) 133–159 (Springer, 2009).
- Likharev, K. K. & Korotkov, A. N. Single-electron parametron: reversible computation in a discrete-state system. *Science* **273**, 763–765 (1996).
- Feynman, R. P. Simulating physics with computers. *Int. J. Theor. Phys.* **21**, 467–488 (1982).
- Siegelmann, H. T. & Sontag, E. D. Analog computation via neural networks. *Theor. Comput. Sci.* **131**, 331–360 (1994).
- Yoshihito, A. Information processing using intelligent materials—information-processing architectures for material processors. *J. Intel. Mater. Syst. Str.* **5**, 418–423 (1994).
- Miller, J. F. & Downing, K. in *Proceedings of the 2002 NASA/DOD Conference on Evolvable Hardware* 167–176 (IEEE, 2002).
- Miller, J. F., Harding, S. L. & Tufte, G. Evolution-in-materio: evolving computation in materials. *Evol. Intel.* **7**, 49–67 (2014).
- Tour, J. M. *et al.* Nanocell logic gates for molecular computing. *IEEE Trans. Nanotechnol.* **1**, 100–109 (2002).
- Chen, R. H., Korotkov, A. N., & Likharev, K. K. Single-electron transistor logic. *Appl. Phys. Lett.* **68**, 1954–1956 (1996).
- Nakajima, F., Miyoshi, Y., Motohisa, J. & Fukui, T. Single-electron AND/NAND logic circuits based on a self-organized dot network. *Appl. Phys. Lett.* **83**, 2680–2682 (2003).
- Maeda, K. *et al.* Logic operations of chemically assembled single-electron transistor. *ACS Nano* **6**, 2798–2803 (2012).
- Heath, J. R., Kuekes, P. J., Snider, G. S. & Williams, R. S. A defect-tolerant computer architecture: opportunities for nanotechnology. *Science* **280**, 1716–1721 (1998).
- Snider, G. S. & Williams, R. S. Nano/CMOS architectures using a field-programmable nanowire interconnect. *Nanotechnology* **18**, 035204 (2007).
- Merolla, P. A. *et al.* A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **345**, 668–673 (2014).
- Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**, 671–680 (1983).
- Holland, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (Univ. Michigan Press, 1975).
- Valiant, L. G. Evolvability. *JACM* **56**, 3 (2009).
- Rodriguez-Vázquez, A. *et al.* ACE16k: the third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs. *IEEE Trans. Circuits Syst. I* **51**, 851–863 (2004).

Acknowledgements

We thank A.-J. Annema, M. Danish, J. Huskens, S. Intan, M. de Jong, J. Mikhail, B. Nauta, D. Reinhoudt, I. Rianasari, E. Strambini, F. Zwanenburg and all the collaborators of the NASCENCE project for fruitful discussions. We acknowledge financial support from MESA+, CTIT, the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement No. 317662 and the European Research Council, ERC Starting Grant No. 240433.

Author contributions

S.K.B. and C.P.L. fabricated the samples, carried out the experiments and performed the data analysis. C.P.L. designed and programmed the genetic search algorithm. R.M.J.v.D. contributed with theoretical inputs. W.G.v.d.W. conceived the experiments, and planned and supervised the project. H.J.B. conceived the project together with W.G.v.d.W. and cosupervised. Z.L. and K.S.M. contributed to the sample fabrication. All the authors discussed the results, provided important insights and helped write the manuscript.

Additional information

Supplementary information is available in the [online version](#) of the paper. Reprints and permissions information is available online at www.nature.com/reprints. Correspondence and requests for materials should be addressed to W.G.v.d.W.

Competing financial interests

The authors declare no competing financial interests.

Methods

Chemicals. The NP solution used in this study was purchased from NanoPartz. The solution consists of Au NPs of 20 nm diameter coated with a monolayer of 1-octanethiol ($\text{CH}_3(\text{CH}_2)_7\text{CH}_2\text{SH}$) stabilized in ethanol. This solution was diluted with ethylene glycol (ethane-1,2-diol) and has a high boiling point (197 °C), high viscosity (16 cP) and high dielectric constant (41.4). These properties, combined with a proper contact angle with respect to the SiO_2 surface, facilitate the NP-trapping procedure explained below.

Electrode fabrication. The metal electrodes were fabricated using electron-beam lithography (EBL) followed by directional metal deposition. The substrate is 0.5 mm thick highly doped Si (p++) with a thermally grown 35 nm SiO_2 top layer. This substrate allows the application of a back-gate voltage. After EBL patterning of the 100 nm thick polymethyl methacrylate resist layer, residual organic contaminants were removed by a quick oxygen-plasma treatment. Thereafter, 30 nm Au with a 5 nm Ti adhesion layer was deposited via vertical electron-beam evaporation. Metal lift-off results in 50 nm wide radial electrodes around an open central area of 200 nm (as shown in Fig. 1a). This geometry was chosen to have in the order of ten Au NPs in between the electrodes. Both eight- and twelve-pin geometries were fabricated, but in the present study only eight-pin geometries were used.

Trapping of NPs. The technique employed to fabricate the NP network is dielectrophoresis, whereby a non-uniform electric field applied across the electrodes drives the suspended NPs to the area in between the electrodes^{31,32}. Before the trapping procedure, the Ti/Au electrodes were cleaned with an oxygen-plasma surface treatment followed by an ethanol rinse and dry. The $11 \times 11 \text{ mm}^2$ chip that contained several eight- and twelve-pin geometries was placed inside a probe station, and a drop of Au NPs suspended in ethylene glycol was dispensed on it. The trapping was done sequentially with pairs of diametrically opposed electrodes, by contacting the pads with the probes. The typical optimized parameters for trapping were frequency (f) = 1 MHz, a.c. voltage (V_{ac}) = 4 V_{pp} (pp, peak to peak) and a trapping time (t) of five minutes. After NP trapping, the chip was cleaned in isopropyl alcohol to remove the physisorbed NPs, and dried to leave behind a localized network between the electrodes, as shown in Fig. 2a and Supplementary Fig. 1a. The NP network sticks well to the substrate and is expected to be mechanically stable at the low temperatures of our experiments.

Genetic search algorithm

Genome and genes. We define a genome as a set of voltages (genes) that determine the device performance. Each gene g_j ($j = 1, 2, \dots, 6$) takes a value from 0 to 1 that corresponds to the control voltage V_j . For the electrodes in contact with the NP network, the gene values (0/1) map to the voltage required to flow a certain I_{OUT} ($\pm \sim 500$ pA). This limits the total output current, by which we hope to increase the fraction of functional genomes with a ~ 100 pA output level. For the back gate, the gene values 0/1 map to 0/−2 V to utilize the back-gate tunability (see Supplementary Fig. 1e).

Breeding mechanism. To preserve the best-performing genomes, and yet search for better-performing candidates, a composite cloning–breeding approach was used, as depicted in Supplementary Fig. 2. It consists of a uniform crossover, with each of the six genes in the offspring $G_{(n+1),i}$ assigned randomly from either parent, $G_{n,i}$ or $G_{n,k}$, with a mixing ratio of 0.5, followed by the randomization of every gene with mutation probability $P = 0.1$. After sorting the 20 genomes $G_{n,i}$ of the n th generation, G_m , in order of descending fitness, the top five (‘elite’) genomes, $G_{n,1} - G_{n,5}$, are cloned into $G_{(n+1),1} - G_{(n+1),5}$ of $G_{(n+1)}$. For the next five genomes of $G_{(n+1)}$, the genes of $G_{n,1} - G_{n,5}$ are modified by +1 and −1%, which results in $G_{n,i}^{+1\%}$ and $G_{n,i}^{-1\%}$ ($i = 1 - 5$), after which they are cross bred one by one, $G_{n,i}^{+1\%} \times G_{n,i}^{-1\%}$. The elite parent combinations with $G_{n,1}$ breeding with $G_{n,2}$, $G_{n,2}$ with $G_{n,3}$ and so on, up to $G_{n,5}$ with $G_{n,6}$, produce another five genomes of $G_{(n+1)}$. The remaining five genomes are generated from breeding $G_{n,1} - G_{n,5}$ with a genome randomly taken from $G_{n,1} - G_{n,20}$. When $G_{(n+1)}$ is complete, its genomes $G_{(n+1),j}$ are sorted based on their fitness, after which the above procedure is repeated for the next generation.

Mutation. A gene g_j can mutate into g_j^* with a probability P . If it mutates, g_j^* is randomly picked out of a triangular distribution from 0 to 1 with a mode about g_j . For $P = 0$, the GA may converge to a local maximum determined by the initial generation; for $P = 1$, it does not exploit the breeding mechanism, as every generation is reset to a random generation. For our GA search reported here, we used a value of $P = 0.1$ that was empirically found to give the best results.

Fitness determination procedure

Input waveform sampling. As input signals, we used well-defined voltage-pulse trains $V(t)$. The slow rate of the signal $V(t)$ should take into account the network’s resistor–capacitor times to maintain the shape of the inputs. In our measurement set-up, the slow rate is 15 milliseconds for d.c. lines, and therefore we chose the input-voltage signals to have double the rise time (that is, 30 millisecond). We chose a sampling time of $\tau = 100 \mu\text{s}$ to update the digital-to-analog converter voltages. Hence, the vector $\mathbf{V}[a] = V(a\tau)$ defines our input. In the case of two-input signals, there is a time delay τ^* between settings $V_1[a]$ and $V_2[a]$. However, $\tau^* \ll \tau$ and hence, for all practical purposes, we can consider the inputs to be applied synchronously. In Fig. 2b, $\mathbf{V}[a] = 0/100$ mV corresponds to logic 0/1, and the intermediate voltages during logic transitions correspond to ‘don’t care’ terms.

Output waveform sampling. At the output electrode, the I/V converter (current amplifier) converts current into voltage and the analogue-to-digital converter measures an output $Y[a]$ at a sampling time τ .

Ideal output. We define an ideal output X that follows Supplementary Fig. 4 when inputs are logic 0/1. In the case of intermediate voltages during logic transitions, X is set to interpolated values. To neglect measurements during these ‘don’t care’ intervals, we define weights W as shown in Supplementary Fig. 5.

Linear fitting. We linearly fit between X and Y in the form $Y = mX + c$, with weights W . Here, m is the slope and c is the intercept. 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 chose the bisquare fit, as we assume that in an evolutionary GA it is better to minimize the influence of outliers in deciding the fitness of genomes.

Fitness score. The linear fit calculates a slope m , intercept c and residue r . These parameters quantify the output’s signal-to-noise ratio ($\text{SNR} = m/\sqrt{r}$) and offset to zero (c). Combining these qualities with a mixing parameter δ , we define the fitness score as $F = m/(\sqrt{r} + \delta c)$. With $\delta = 0$, we found genomes with a good SNR but arbitrary offset (especially for negating logic gates). To fulfil the requirements of a good SNR together with a low offset, an empirical value of $\delta = 0.01$ was used for all the measurements reported here.

GA convergence. 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 . As shown in Supplementary Fig. 3, for multiple reruns of the GA search for a NAND gate solution, the GA search converged in 11 out of 12 identical cases.

References

- Pohl, H. A. *Dielectrophoresis: the Behavior of Neutral Matter in Nonuniform Electric Fields* (Cambridge Monographs in Physics, 80, Cambridge Univ. Press, 1978).
- Bernard, L., Calame, M., van der Molen, S., Liao, J. & Schönenberger, C. Controlled formation of metallic nanowires via Au nanoparticle ac trapping. *Nanotechnology* **18**, 235202–235207 (2007).