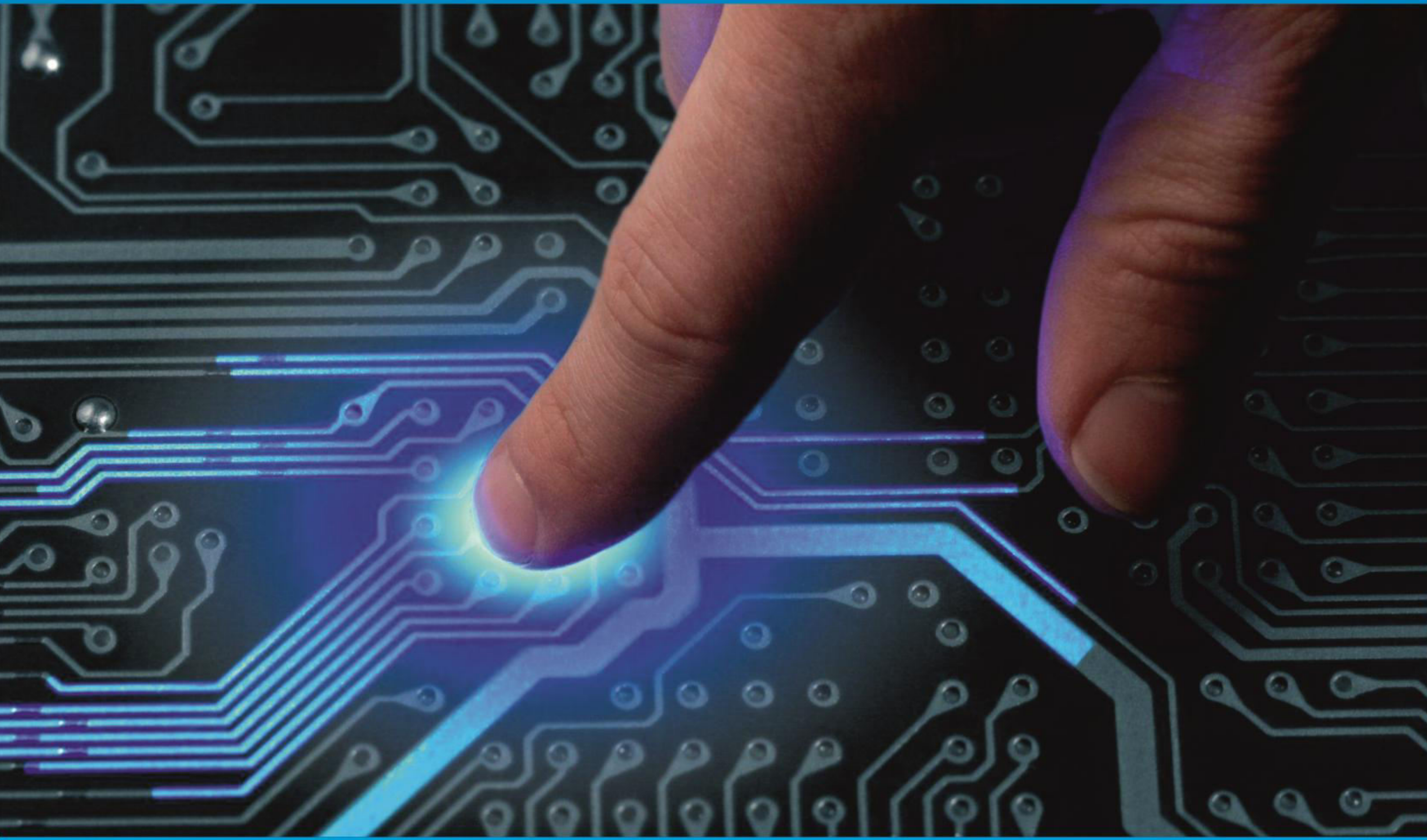
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 1, January 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 9, Issue 1, January 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0901043 |

Chemical Computer- A New Era Wonder

Dr. Anil Kumar

Dept. of Chemistry, D.A.V. [P.G.] College, Dehradun, India

ABSTRACT: A chemical computer, also called a reaction-diffusion computer, Belousov–Zhabotinsky (BZ) computer, or gooware computer, is an unconventional computer based on a semi-solid chemical "soup" where data are represented by varying concentrations of chemicals.[1] The computations are performed by naturally occurring chemical reactions.

Keywords: chemical,computer, unconventional, reactions,computations

I. INTRODUCTION

Originally chemical reactions were seen as a simple move towards a stable equilibrium which was not very promising for computation. This was changed by a discovery made by Boris Belousov, a Soviet scientist, in the 1950s. He created a chemical reaction between different salts and acids that swing back and forth between being yellow and clear because the concentration of the different components changes up and down in a cyclic way. At the time this was considered impossible because it seemed to go against the second law of thermodynamics, which says that in a closed system the entropy will only increase over time, causing the components in the mixture to distribute themselves until equilibrium is gained and making any changes in the concentration impossible. But modern theoretical analyses shows sufficiently complicated reactions can indeed comprise wave phenomena without breaking the laws of nature.[1][2] (A convincing directly visible demonstration was achieved by Anatol Zhabotinsky with the Belousov–Zhabotinsky reaction showing spiraling colored waves.)

The wave properties of the BZ reaction means it can move information in the same way as all other waves. This still leaves the need for computation, performed by conventional microchips using the binary code transmitting and changing ones and zeros through a complicated system of logic gates. To perform any conceivable computation it is sufficient to have NAND gates. (A NAND gate has two bits input. Its output is 0 if both bits are 1, otherwise it's 1). In the chemical computer version logic gates are implemented by concentration waves blocking or amplifying each other in different ways.

Current research

In 1989 it was demonstrated how light-sensitive chemical reactions could perform image processing.[3] This led to an upsurge in the field of chemical computing. Andrew Adamatzky at the University of the West of England has demonstrated simple logic gates using reaction–diffusion processes.[4] Furthermore, he has theoretically shown how a hypothetical "2^+ medium" modelled as a cellular automaton can perform computation.[5] Adamatzky was inspired by a theoretical article on computation by using balls on a billiard table to transfer this principle to the BZ-chemicals and replace the billiard balls with waves: if two waves meet in the solution, they create a third wave which is registered as a 1. He has tested the theory in practice and is working to produce some thousand chemical versions of logic gates to create a chemical pocket calculator. One of the problems with the present version of this technology is the speed of the waves; they only spread at a rate of a few millimeters per minute. According to Adamatzky, this problem can be eliminated by placing the gates very close to each other, to make sure the signals are transferred quickly. Another possibility could be new chemical reactions where waves propagate much faster.

In 2014, a chemical computing system was developed by an international team headed by the Swiss Federal Laboratories for Materials Science and Technology (Empa). The chemical computer used surface tension calculations derived from the Marangoni effect using an acidic gel to find the most efficient route between points A and B, outpacing a conventional satellite navigation system attempting to calculate the same route.[6][7]

In 2015, Stanford University graduate students created a computer using magnetic fields and water droplets infused with magnetic nanoparticles, illustrating some of the basic principles behind a chemical computer.[8][9]

In 2015, University of Washington students created a programming language for chemical reactions (originally developed for DNA analysis).[10][11]

In 2017, researchers at Harvard University patented a chemical Turing machine that operated using the non-linear dynamics of the Belousov–Zhabotinsky reaction.[12] The system that they developed is capable of recognizing a Chomsky type-1 language using Gibbs free energy considerations. This work was subsequently published in 2019, including systems for Chomsky type-2 and type-3 languages.[13]

In 2020, University of Glasgow researchers created a chemical computer using 3D-printed parts and magnetic stirrers in order to control the oscillations of BZ medium. In doing so, they were able to compute binary logic gates, and perform pattern recognition.[14]

II. DISCUSSION

Biological computers use biologically derived molecules — such as DNA and/or proteins — to perform digital or real computations.

The development of biocomputers has been made possible by the expanding new science of nanobiotechnology. The term nanobiotechnology can be defined in multiple ways; in a more general sense, nanobiotechnology can be defined as any type of technology that uses both nano-scale materials (i.e. materials having characteristic dimensions of 1-100 nanometers) and biologically based materials.[1] A more restrictive definition views nanobiotechnology more specifically as the design and engineering of proteins that can then be assembled into larger, functional structures[2][3] The implementation of nanobiotechnology, as defined in this narrower sense, provides scientists with the ability to engineer biomolecular systems specifically so that they interact in a fashion that can ultimately result in the computational functionality of a computer.

Biocomputers use biologically derived materials to perform computational functions. A biocomputer consists of a pathway or series of metabolic pathways involving biological materials that are engineered to behave in a certain manner based upon the conditions (input) of the system. The resulting pathway of reactions that takes place constitutes an output, which is based on the engineering design of the biocomputer and can be interpreted as a form of computational analysis. Three distinguishable types of biocomputers include biochemical computers, biomechanical computers, and bioelectronic computers.[4]

Biochemical computers

Biochemical computers use the immense variety of feedback loops that are characteristic of biological chemical reactions in order to achieve computational functionality.[5] Feedback loops in biological systems take many forms, and many different factors can provide both positive and negative feedback to a particular biochemical process, causing either an increase in chemical output or a decrease in chemical output, respectively. Such factors may include the quantity of catalytic enzymes present, the amount of reactants present, the amount of products present, and the presence of molecules that bind to and thus alter the chemical reactivity of any of the aforementioned factors. Given the nature of these biochemical systems to be regulated through many different mechanisms, one can engineer a chemical pathway comprising a set of molecular components that react to produce one particular product under one set of specific chemical conditions and another particular product under another set of conditions. The presence of the particular product that results from the pathway can serve as a signal, which can be interpreted—along with other chemical signals—as a computational output based upon the starting chemical conditions of the system (the input).

Biomechanical computers

Biomechanical computers are similar to biochemical computers in that they both perform a specific operation that can be interpreted as a functional computation based upon specific initial conditions which serve as input. They differ, however, in what exactly serves as the output signal. In biochemical computers, the presence or concentration of certain chemicals serves as the input signal. In biomechanical computers, however, the mechanical shape of a specific molecule or set of molecules under a set of initial conditions serves as the output. Biomechanical computers rely on the nature of specific molecules to adopt certain physical configurations under certain chemical conditions. The mechanical, three-dimensional structure of the product of the biomechanical computer is detected and interpreted appropriately as a calculated output.

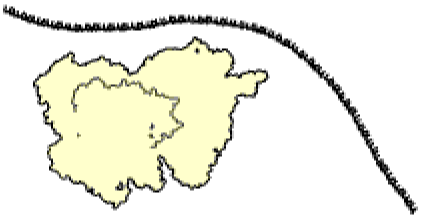
Bioelectronic computers

Biocomputers can also be constructed in order to perform electronic computing. Again, like both biomechanical and biochemical computers, computations are performed by interpreting a specific output that is based upon an initial set of conditions that serve as input. In bioelectronic computers, the measured output is the nature of the electrical conductivity that is observed in the bioelectronic computer. This output comprises specifically designed biomolecules that conduct electricity in highly specific manners based upon the initial conditions that serve as the input of the bioelectronic system.

Network-based biocomputers

In networks-based biocomputation,[6] self-propelled biological agents, such as molecular motor proteins or bacteria, explore a microscopic network that encodes a mathematical problem of interest. The paths of the agents through the network and/or their final positions represent potential solutions to the problem. For instance, in the system described by Nicolau et al.,[6] mobile molecular motor filaments are detected at the "exits" of a network encoding the NP-complete problem SUBSET SUM. All exits visited by filaments represent correct solutions to the algorithm. Exits not visited are non-solutions. The motility proteins are either actin and myosin or kinesin and microtubules. The myosin and kinesin, respectively, are attached to the bottom of the network channels. When adenosine triphosphate (ATP) is added, the actin filaments or microtubules are propelled through the channels, thus exploring the network. The energy conversion from chemical energy (ATP) to mechanical energy (motility) is highly efficient when compared with e.g. electronic computing, so the computer, in addition to being massively parallel, also uses orders of magnitude less energy per computational step.

Engineering biocomputers



A ribosome is a biological machine that uses protein dynamics on nanoscales to translate RNA into proteins

The behavior of biologically derived computational systems such as these relies on the particular molecules that make up the system, which are primarily proteins but may also include DNA molecules. Nanobiotechnology provides the means to synthesize the multiple chemical components necessary to create such a system.[citation needed] The chemical nature of a protein is dictated by its sequence of amino acids—the chemical building blocks of proteins. This sequence is in turn dictated by a specific sequence of DNA nucleotides—the building blocks of DNA molecules. Proteins are manufactured in biological systems through the translation of nucleotide sequences by biological molecules called ribosomes, which assemble individual amino acids into polypeptides that form functional proteins based on the nucleotide sequence that the ribosome interprets. What this ultimately means is that one can engineer the chemical components necessary to create a biological system capable of performing computations by engineering DNA nucleotide sequences to encode for the necessary protein components. Also, the synthetically designed DNA molecules themselves may function in a particular biocomputer system. Thus, implementing nanobiotechnology to design and produce synthetically designed proteins—as well as the design and synthesis of artificial DNA molecules—can allow the construction of functional biocomputers (e.g. Computational Genes).

Biocomputers can also be designed with cells as their basic components. Chemically induced dimerization systems can be used to make logic gates from individual cells. These logic gates are activated by chemical agents that induce interactions between previously non-interacting proteins and trigger some observable change in the cell.[7]

Network-based biocomputers are engineered by nanofabrication of the hardware from wafers where the channels are etched by electron-beam lithography or nano-imprint lithography. The channels are designed to have a high aspect ratio of cross section so the protein filaments will be guided. Also, split and pass junctions are engineered so filaments will propagate in the network and explore the allowed paths. Surface silanization ensures that the motility proteins can be affixed to the surface and remain functional. The molecules that perform the logic operations are derived from biological tissue.

Economics

All biological organisms have the ability to self-replicate and self-assemble into functional components. The economical benefit of biocomputers lies in this potential of all biologically derived systems to self-replicate and self-assemble given appropriate conditions.[4]:349 For instance, all of the necessary proteins for a certain biochemical pathway, which could be modified to serve as a biocomputer, could be synthesized many times over inside a biological cell from a single DNA molecule. This DNA molecule could then be replicated many times over. This characteristic of biological molecules could make their production highly efficient and relatively inexpensive. Whereas electronic computers require manual production, biocomputers could be produced in large quantities from cultures without any additional machinery needed to assemble them.

Notable advancements in biocomputer technology

Currently, biocomputers exist with various functional capabilities that include operations of "binary " logic and mathematical calculations.[5] Tom Knight of the MIT Artificial Intelligence Laboratory first suggested a biochemical computing scheme in which protein concentrations are used as binary signals that ultimately serve to perform logical operations.[4]:349 At or above a certain concentration of a particular biochemical product in a biocomputer chemical pathway indicates a signal that is either a 1 or a 0. A concentration below this level indicates the other, remaining signal. Using this method as computational analysis, biochemical computers can perform logical operations in which the appropriate binary output will occur only under specific logical constraints on the initial conditions. In other words, the appropriate binary output serves as a logically derived conclusion from a set of initial conditions that serve as premises from which the logical conclusion can be made. In addition to these types of logical operations, biocomputers have also been shown to demonstrate other functional capabilities, such as mathematical computations. One such example was provided by W.L. Ditto, who in 1999 created a biocomputer composed of leech neurons at Georgia Tech which was capable of performing simple addition.[4]:351 These are just a few of the notable uses that biocomputers have already been engineered to perform, and the capabilities of biocomputers are becoming increasingly sophisticated. Because of the availability and potential economic efficiency associated with producing biomolecules and biocomputers—as noted above—the advancement of the technology of biocomputers is a popular, rapidly growing subject of research that is likely to see much progress in the future.

In March 2013. a team of bioengineers from Stanford University, led by Drew Endy, announced that they had created the biological equivalent of a transistor, which they dubbed a "transcriptor". The invention was the final of the three components necessary to build a fully functional computer: data storage, information transmission, and a basic system of logic.[8]

In July 2017, separate experiments with E. Coli published on Nature showed the potential of using living cells for computing tasks and storing information. A team formed with collaborators of the Biodesign Institute at Arizona State University and Harvard's Wyss Institute for Biologically Inspired Engineering developed a biological computer inside E. Coli that responded to a dozen inputs. The team called the computer "ribocomputer", as it was composed of ribonucleic acid. Harvard researchers proved that it is possible to store information in bacteria after successfully archiving images and movies in the DNA of living E. coli cells.[9]

In 2020, a team led by biophysicist Sangram Bagh realized a study with E. coli to solve 2 x 2 maze problems to probe the principle for distributed computing among cells.[10][11]

Parallel biological computing with networks, where bio-agent movement corresponds to arithmetical addition was demonstrated in 2016 on a SUBSET SUM instance with 8 candidate solutions.[6]

Future potential of biocomputers

Many examples of simple biocomputers have been designed, but the capabilities of these biocomputers are very limited in comparison to commercially available non-bio computers. Some people believe that biocomputers have great potential, but this has yet to be demonstrated. The potential to solve complex mathematical problems using far less energy than standard electronic supercomputers, as well as to perform more reliable calculations simultaneously rather than sequentially, motivates the further development of "scalable" biological computers, and several funding agencies are supporting these efforts.[12][13]

III. RESULTS

Organic computing is computing that behaves and interacts with humans in an organic manner. The term "organic" is used to describe the system's behavior, and does not imply that they are constructed from organic materials. It is based on the insight that we will soon be surrounded by large collections of autonomous systems, which are equipped with sensors and actuators, aware of their environment, communicate freely, and organize themselves in order to perform the actions and services that seem to be required.

The goal is to construct such systems as robust, safe, flexible, and trustworthy as possible. In particular, a strong orientation towards human needs as opposed to a pure implementation of the technologically possible seems absolutely central. In order to achieve these goals, our technical systems will have to act more independently, flexibly, and autonomously, i.e. they will have to exhibit lifelike properties. We call such systems "organic". Hence, an "Organic Computing System" is a technical system which adapts dynamically to exogenous and endogenous change. It is characterized by the properties of self-organization, self-configuration, self-optimization, self-healing, self-protection, self-explaining, and context awareness. It can be seen as an extension of the Autonomic computing vision of IBM.

In a variety of research projects the priority research program SPP 1183 of the German Research Foundation (DFG) addresses fundamental challenges in the design of Organic Computing systems; its objective is a deeper understanding of emergent global behavior in self-organizing systems and the design of specific concepts and tools to support the construction of Organic Computing systems for technical applications.

A quantum computer is a computer that exploits quantum mechanical phenomena. At small scales, physical matter exhibits properties of both particles and waves, and quantum computing leverages this behavior, specifically quantum superposition and entanglement, using specialized hardware that supports the preparation and manipulation of quantum states. Classical physics cannot explain the operation of these quantum devices, and a scalable quantum computer could perform some calculations exponentially faster than any modern "classical" computer. In particular, a large-scale quantum computer could break widely used encryption schemes and aid physicists in performing physical simulations; however, the current state of the art is largely experimental and impractical, with several obstacles to useful applications.

The basic unit of information in quantum computing is the qubit, similar to the bit in traditional digital electronics. Unlike a classical bit, a qubit can exist in a superposition of its two "basis" states, which loosely means that it is in both states simultaneously. When measuring a qubit, the result is a probabilistic output of a classical bit. If a quantum computer manipulates the qubit in a particular way, wave interference effects can amplify the desired measurement results. The design of quantum algorithms involves creating procedures that allow a quantum computer to perform calculations efficiently and quickly.

Physically engineering high-quality qubits has proven challenging. If a physical qubit is not sufficiently isolated from its environment, it suffers from quantum decoherence, introducing noise into calculations. National governments have invested heavily in experimental research that aims to develop scalable qubits with longer coherence times and lower error rates. Two of the most promising technologies are superconductors (which isolate an electrical current by eliminating electrical resistance) and ion traps (which confine a single ion using electromagnetic fields).

In principle, a non-quantum (classical) computer can solve the same computational problems as a quantum computer, given enough time. Quantum advantage comes in the form of time complexity rather than computability, and quantum complexity theory shows that some quantum algorithms for carefully selected tasks require exponentially fewer computational steps than the best known non-quantum algorithms. Such tasks can in theory be solved on a large-scale quantum computer whereas classical computers would not finish computations in any reasonable amount of time. However, quantum speedup is not universal or even typical across computational tasks, since basic tasks such as sorting are proven to not allow any asymptotic quantum speedup. Claims of quantum supremacy have drawn significant attention to the discipline, but are demonstrated on contrived tasks, while near-term practical use cases remain limited.

Optimism about quantum computing is fueled by a broad range of new theoretical hardware possibilities facilitated by quantum physics, but the improving understanding of quantum computing limitations counterbalances this optimism. In particular, quantum speedups have been traditionally estimated for noiseless quantum computers, whereas the impact of noise and the use of quantum error-correction can undermine low-polynomial speedups.

IV. CONCLUSIONS

Any computational problem solvable by a classical computer is also solvable by a quantum computer.[134] Intuitively, this is because it is believed that all physical phenomena, including the operation of classical computers, can be described using quantum mechanics, which underlies the operation of quantum computers.

Conversely, any problem solvable by a quantum computer is also solvable by a classical computer. It is possible to simulate both quantum and classical computers manually with just some paper and a pen, if given enough time. More formally, any quantum computer can be simulated by a Turing machine. In other words, quantum computers provide no additional power over classical computers in terms of computability. This means that quantum computers cannot solve undecidable problems like the halting problem, and the existence of quantum computers does not disprove the Church–Turing thesis.

REFERENCES

1. Kumar, Ambar; Mahato, Akash Kumar; Singh, Akashdeep (2014). "Chemical Computing: The different way of computing" (PDF). International Journal of Innovative Research in Technology. 1 (6). ISSN 2349-6002. Archived from the original (PDF) on 2015-06-15. Retrieved 2015-06-14.
2. ^ "Moore's Law Is About to Get Weird". Nautilus.
3. ^ L. Kuhnert; K. I. Agladze; V. I. Krinsky (1989). "Image processing using light-sensitive chemical waves". Nature. 337 (6204): 244–247. Bibcode:1989Natur.337..244K. doi:10.1038/337244a0. S2CID 4257968.
4. ^ Adamatzky, Andrew; De Lacy Costello, Benjamin (2002). "Experimental logical gates in a reaction-diffusion medium: The XOR gate and beyond". Physical Review E. 66 (4): 046112. Bibcode:2002PhRvE..66d6112A. doi:10.1103/PhysRevE.66.046112. PMID 12443264.
5. ^ Andrew I. Adamatzky (1997). "Information-processing capabilities of chemical reaction-diffusion systems. 1. Belousov-Zhabotinsky media in hydrogel matrices and on solid supports". Advanced Materials for Optics and Electronics. 7 (5): 263–272. doi:10.1002/(SICI)1099-0712(199709)7:5<263::AID-AMO317>3.0.CO;2-Y.
6. ^ "Chemical GPS Outpreforms Satellite Navigation System > ENGINEERING.com". engineering.com.
7. ^ "Empa invents chemical computer faster than a satnav". gizmag.com. 28 October 2014.
8. ^ "Stanford has created a water-droplet computer - ExtremeTech". ExtremeTech.
9. ^ "This computer clocks uses water droplets, manipulating information and matter at the same time". ZME Science. 10 June 2015.
10. ^ Taylor Soper (30 September 2013). "Chemical computer: Researchers develop programming language to control DNA molecules". GeekWire.
11. ^ "UW engineers invent programming language to build synthetic DNA". washington.edu.
12. ^ US 9 582 771
13. ^ Dueñas-Díez M, Pérez-Mercader J (2019). "How Chemistry Computes: Language Recognition by Non-Biochemical Chemical Automata. From Finite Automata to Turing Machines". iScience. 19: 514–526. Bibcode:2019iSci...19..514D. doi:10.1016/j.isci.2019.08.007. PMC 6710637. PMID 31442667.
14. ^ Parrilla-Gutierrez, Juan Manuel; Sharma, Abhishek; Tsuda, Soichiro; Cooper, Geoffrey J. T.; Aragon-Camarasa, Gerardo; Donkers, Kevin; Cronin, Leroy (2020). "A programmable chemical computer with memory and pattern recognition". Nature Communications. 11 (1): 1442. Bibcode:2020NatCo..11.1442P. doi:10.1038/s41467-020-15190-3. PMC 7080730. PMID 32188858.
15. Wispelway. June. "Nanobiotechnology: The Integration of Nanoengineering and Biotechnology to the Benefit of Both." Society for Biological Engineering (Special Section): Nanobiotechnology, p. 34
16. ^ Ratner. Daniel and Mark. Nanotechnology: A Gentle Introduction to the Next Big Idea. Pearson Education. Inc: 2003, p. 116-7
17. ^ Gary Stix. "Little Big Science." Understanding Nanotechnology (p6-16). Scientific American. Inc. and Byron Preiss Visual Publications. Inc: 2002, p. 9
18. ^ [a b c d] Freitas. Robert A. Nanomedicine Volume I: Basic Capabilities. Austin. Texas: Landes Bioscience. 1999.[:349–51]
19. ^ [a b] Windmiller, Joshua (June 2012). Molecular scale biocomputing : an enzyme logic approach (Thesis). UC San Diego.
20. ^ [a b c] Nicolau, Dan V.; Lard, Mercy; Korten, Till; van Delft, Falco C. M. J. M.; Persson, Malin; Bengtsson, Elina; Månsson, Alf; Diez, Stefan; Linke, Heiner; Nicolau, Dan V. (8 March 2016). "Parallel computation with molecular-motor-propelled agents in nanofabricated networks". Proceedings of the National Academy of Sciences. 113 (10): 2591–2596. Bibcode:2016PNAS..113.2591N. doi:10.1073/pnas.1510825113. PMC 4791004. PMID 26903637.
21. ^ Miyamoto, T; DeRose. R; Suarez. A; Ueno. T; Chen. M; Sun. TP; Wolfgang. MJ; Mukherjee. C; Meyers. DJ; Inoue. T (Mar 25, 2012). "Rapid and orthogonal logic gating with a gibberellin-induced dimerization system". Nature Chemical Biology. 8 (5): 465–70. doi:10.1038/nchembio.922. PMC 3368803. PMID 22446836.

22. ^ Robert T. Gonzalez (March 29, 2013). "This new discovery will finally allow us to build biological computers". IO9. Retrieved March 29, 2013.
23. ^ Waltz, Emily (12 July 2017). "Scientists Store Video Data in the DNA of Living Organisms". IEEE Spectrum. Retrieved 28 November 2020.
24. ^ Sarkar, Kathakali; Chakraborty, Saswata; Bonnerjee, Deepro; Bagh, Sangram (15 October 2020). "Distributed Computing with Engineered Bacteria and Its Application in Solving Chemically Generated 2 × 2 Maze Problems". ACS Synthetic Biology. 10 (10): 2456–2464. doi:10.1021/acssynbio.1c00279.
25. ^ Siobhan Roberts, Siobhan. "An E. coli biocomputer solves a maze by sharing the work". MIT Technology Review. Retrieved 27 November 2020.
26. ^ "Bio4Comp - Parallel Network-based Biocomputation". Bio4Comp Research Project. Retrieved 19 December 2019.
27. ^ Technology (QUT), Queensland University of. "QUT ARC Future Fellowships announced". QUT.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com