

# The role of alternative computing paradigms in engineering research 41712



**ASSIGN  
BUSTER**

## **Abstract**

In the history of computer technology, the physical realization of a computer has always been a

matter of concern and research, because all theoretical computational models are of use only if

they can be readily and feasibly translated into a physical implementation.

There can be no doubt that the most popular paradigm for the physical implementation of a computer has been the

ubiquitous silicon chip, mainly due to its speed and fault-tolerance. However, it comes with its share of shortcomings. In order to overcome the imminent watershed in the field of computing infrastructure, many alternative ways and means of performing computations have been developed. Notable amongst such paradigms are quantum computing [1, 2] and biological computing [3, 4, 5]. These paradigms do not suffer from the major constraint that an emphasis on miniaturization of the processing element places. Some of these paradigms also offer a much higher data density than current silicon data storage devices. Also, they are mostly non-toxic unlike silicon computing. Some of these have grown to become full-fledged computing paradigms, which may in the near future challenge the might of the omnipresent silicon paradigm. In this paper, we review the major computing paradigms, which may serve as alternatives to silicon computing.

**Keywords**

Computing paradigms, Silicon computing, Quantum computing, DNA computing.

## 1. Introduction.

From the days of G. Moore, the inventor of the transistor to the present times of nanotechnological breakthroughs, the silicon chip has evolved to incorporate massive miniaturization. Today, advanced lithographic techniques are widely used to fabricate chips with logic gates and wires whose dimensions are less than a micron. As a result, the number of transistors on a die is of the order of  $10^7$

So far, Moore's prediction that the number of transistors on a chip will increase at an exponential rate has held true. However, the silicon industry has reached its zenith for miniaturization. Beyond the nanotechnological level, the laws of physics prevent us from increasing the computational power of a chip. The classical laws applicable so far must be

replaced by the quantum laws, which propound a particle position uncertainty under which any

consideration of information is susceptible to be uncertain.

## 2. Drawbacks of Silicon

Computing.

The processing power of silicon-based computing can be increased up to the point of the

limitations imposed on it by the laws of physics. The 'processing power' of computers is usually

measured by speed. Here speed means how fast circuits can move information from A to B and how

fast the information can be processed once it gets to B. The traditional computing design paradigm focuses on decreasing the distance that information (in the form of electrical signals) have to travel; in 2 other words, shortening the distance between A and

B. This has meant packing more and more processing elements or transistors into the central

processing chip of the computer. Each of these transistors is essentially a tiny binary on/off switch.

Today, this packing of transistors has reached an amazing level of density. For instance, a common Pentium IV chip packs 55 million transistors in a space the size of a dime. By relentlessly pursuing this miniaturization strategy, computing technology has advanced rapidly in a comparatively short amount of time. To put the advances in perspective, compare the Pentium chip equipped desktop with the ENIAC computer of the 1940's. That device, with its 17, 000 vacuum tubes (the transistor's predecessor) weighed 30 tons and filled an entire room. Yet, its processing power was less than one hundred thousandth that of

the Pentium. As relentless as this progress has been thus far, it is dependent upon continuing to find new means to shrink transistors so as to fit ever  
<https://assignbuster.com/the-role-of-alternative-computing-paradigms-in-engineering-research-41712/>

more of them onto a chip. In the coming years, transistors will have decreased in size to such a great extent that the only

way to make them smaller is to construct them out of individual atoms or small groupings of atoms. Unfortunately, quantum effects of physics operating on that size and scale will prevent the effective transmission of signals. The Heisenberg Uncertainty Principle [6] states that particles (such as the electrons that make up the information signals that flow through computers) can exhibit the strange behavior of being in other places than where they should be. Researchers can never be completely certain where these particles are at any given moment. This means that electrons, which should ideally be speeding down the atomic-scale circuit pathways in future silicon computers, might be someplace else along with the information they were

assigned to carry. Also there are additional problems with silicon-based technology that make finding viable alternatives even more imperative. First, the components out of which computer processing chips are made are toxic, e. g. arsenic, and therefore present challenges in both fabrication and disposal. Second, silicon-based computers are not very energy efficient. They waste a great deal of energy in the form of the heat that they generate and the energy they consume.

With these limitations in mind, let's look at some alternatives to the current computing

paradigm.

<https://assignbuster.com/the-role-of-alternative-computing-paradigms-in-engineering-research-41712/>

### 3. Quantum Computing.

The alternative paradigm of Quantum Computing was born when Feynman realized a simulation of quantum-mechanical objects by other quantum systems in 1982 [1]. However the theoretical foundations of Quantum Computing were laid when Deutsch of the University of Oxford described a theoretical prototype of a Quantum Computer in [7].

The Quantum Computer emerged from the dregs of oblivion to capture the attention of computer scientists when in 1994, Shor devised the first quantum algorithm for factorization [8] –

something that a quantum computer could do much better than any other form of computer.

The suggestion to use an atom as a bit led to the emergence of quantum computing. Apart from

the two distinct electronic states, viz. the excited state and the ground state, the atom can be prepared to be in both the states simultaneously. The act of being in both the states simultaneously is known as coherent superposition.

The idea of a quantum bit (qubit) is based on this point [9].

A mathematical function called as the wave function represents the particular state of the

quantum mechanical system. It is a complex exponential and includes all possible phases of

existence of the state being represented by it. If  $\psi_1$  and  $\psi_2$  are the wave functions of any two

independent states of the system, then according to quantum mechanics, there exists a state of the same system that can be represented by the wave function  $c_1\psi_1 + c_2\psi_2$ . This state is called as the superposition of the two states.

All superpositions of two quantum states of a system need not be stable. A coherent superposition

is one which is stable and this stability is achieved by some sort of coherence between the two states being used for superposition. This concept of coherent superposition is employed in quantum computing. For instance, in a coherent-superpositioned register, operations can be performed on all the numbers simultaneously. 3

The operation performed by a quantum computer in one computational step on  $2^L$  different

input numbers encoded in coherent superpositions of  $L$  qubits is equivalent to the  $2^L$  computational steps of classical computer. Thus, an enormous gain in the use of computational resources is achieved.

The real power of quantum computation lies in new quantum algorithms which allow exploiting

quantum superposition that can contain an exponential number of different terms. A quantum

computer has not yet been realized due to the fact that the technical obstacles are yet to be solved. The ideas and concepts of quantum computing has been demonstrated using methods like Nuclear Magnetic Resonance [10], Ion Trap [11], Quantum Dot [12], Optical Methods and others.

Among the applications of quantum computing cryptography [13], searching, factorizing large numbers very rapidly [14] and simulating quantum-mechanical systems efficiently are the most

important ones. There have been numerous promising advancements in the field of quantum computing since its conception, including the building of two- and three-qubit quantum computers

capable of simple arithmetic and data sorting. In the present state, quantum computing

remains in its pioneering stage and quantum hardware remains an emerging field. But the work

done and recent advancements in the field of quantum computing suggests that it will only be a

matter time we will have devices large enough to test Shor's and other quantum algorithms. Thus, quantum computers will emerge as the superior computational devices. Since quantum computation has its origins in highly specialized fields of theoretical physics, its future undoubtedly lies in the



profound effect it will have on the lives of all mankind. Experimental as well as theoretical research in quantum computation is accelerating world-wide.

#### 4. Biological Computing.

Biological computing is the use of living organisms or their component parts to perform computing operations or operations associated with computing, e.g. storage [15]. The various forms of biological computing take a different route than those used by quantum or optical computing to overcome the limitations to performance that silicon-based computers face. Rather than focusing on increasing the speed of individual computing operations, biological computing focuses on the use of massive parallelism, or the allocation of tiny portions of a computing task to many different processing elements [16]. Each element in and of itself cannot perform its task quickly, but the fact that there is an incredibly huge number of such elements, each performing a small task, means that the processing operation can be performed far more quickly.

Silicon-based computers have used massively parallel processing but will never be capable of the level of massively parallel processing that biological computers can demonstrate. The biological nature of the operation of biological computing also makes it uniquely suited to controlling processes that would require an interface between other biological processes and human technology.

##### 4. 1 DNA Computers.

DNA computers use strands of DNA (deoxyribonucleic acid) to perform computing

operations. DNA, which encodes the design specifications for living things, is composed of four

amino acids: Adenine, Cytosine, Guanine, and Thymine. In DNA, Adenine will only chemically bind to Thymine and Cytosine will only bind to Guanine. This regular, predictable binding behavior

allows DNA strands to arrange themselves into the famous double helix of complete DNA molecules;

this particular characteristic also makes DNA well suited to computation.

DNA computers process information by making and breaking the chemical bonds between

the DNA components. DNA computing involves using a strand of DNA to represent a problem in

math or logic. The structure of the DNA allows the elements of the problem to be represented in a form that is analogous to the binary code structure (1's and 0's) that characterizes the most basic form of computer language.

Trillions of unique strands of DNA are able to represent all of the possible solutions to a problem. The 'possible solution' strands are allowed to react with the 'problem' strands. Given the binding rules for DNA's components, the strands that complement one another will bind, yielding a collection of whole DNA molecules that contain the solutions [17]. A number of processing

steps are performed on the resulting mixture to isolate a correct solution from all the possibilities. The results are then analyzed by the electronic portion of the computer.

The ability of trillions of reactions to occur simultaneously provides the equivalent of massively parallel processing in silicon-based computers, in which a huge number of possible problem solutions or searches through pools of information for the

answer are performed at the same time. By their mode of functioning, these DNA computers can be

applied to solving a variety of problems, just as the traditional silicon-based computers can. This ability to be a general purpose tool that can be programmed to handle various tasks means that DNA computers meet the accepted standard for being classed as a true computer according to the Turing machine concept developed by Turing [18] in 1936.

DNA computing has already begun to prove its worth by solving complicated logic problems

[19]. The first major development came in 1994, when Adleman [20] was able to create and use a

DNA computer to solve what is known as the 'traveling salesman' problem. This involved using

DNA strands to pick the most direct route by which a salesman would visit each of seven cities while passing through each of the cities only once.

<https://assignbuster.com/the-role-of-alternative-computing-paradigms-in-engineering-research-41712/>

Though this is something that a human being could do in a relatively short time with a piece of scratch paper and a pencil, it was an important proof of concept.

That was just the start.

In March, 2002, NASA announced that a team led by the L. Adleman developed a DNA computer

that solved a problem that required evaluating one million alternatives for their ability to meet 24 separate criteria. Work, such as that done by Adleman, is aimed at increasing the complexity of problems [19] that such computers can solve, while also developing measures to reduce the levels of errors in the process.

Not all DNA computing adheres to the format established by Adleman's work. A variant of DNA

computing is being designed to emulate more closely the structure of conventional computers.

Scientists at the University of Rochester in 1997 developed logic gates made of DNA [11]. Instead of using electrical signals to perform logical operations, DNA logic gates rely on inputs and outputs of DNA. They are designed to detect fragments of genetic material and splice the fragments together to form a single output[21].

In November, 2001, scientists in Israel developed a series of tiny DNA computers, trillions

of which can fit in solution in a test tube. These computers use a form of software, also made of DNA, and can compute with high reliability [22].

#### 4. 2 Genetic Algorithms.

A genetic algorithm is any type of software that uses variation and selection to produce an

evolutionarily tuned output [23]. That output may be a program, a value, or even a picture. The genetic algorithm needs a process for generating new variants and feedback, or fitness criteria, in order to determine which variants to discard and which to use[24]. In a genetic algorithm, the problem is encoded in a series of bit strings that are manipulated by the algorithm.

The genetic algorithm is often mentioned in connection with Artificial Life (Alife) [25], which is the study of virtual organisms created in a computer.

These virtual organisms usually live on a virtual grid, and sometimes even reproduce with each other and consume virtual nutrients. These Alife simulations are occasionally used to study real life forms, or to evolve behaviors shared with real life forms. By creating our own artificial evolutionary environments using genetic algorithms, we can view types of life that previously existed only in the imagination.

In one Alife simulation, 'organisms' are merely lit-up pixels, which influence pixels around

them based on whether adjacent pixels are lit up or not[26]. Surprisingly, the interactions between these organisms can create complex high-level effects that no one could have predicted by looking at the low-level components.

Because genetic algorithms may explore large portions of the search space to a particular problem, they may yield solutions that human operators never would have thought up. The downside to these programs is that for problems with a very large search space, the number of possible solutions may be huge. Therefore, the system can be computationally intensive. By uniting the power of human thought with the brute search-and-test randomness of genetic algorithms, we can create unique new solutions to problems in biology, engineering, mathematics, and other fields. 5

#### 4. 3 Genetic Programs and Genetic

Robots.

Researchers are developing genetic ‘ computer programs’ that could be introduced into and

replicated by living cells in order to control their processes. Research has already produced engineered sequences of genetic material that can cause the living cell in which it is implanted to produce one of two possible genes [27]. This would be effectively analogous to computer programs. This technology could serve as ‘ switches’ to control the chemicals that living organisms synthesize [28]. Development efforts are underway to add data processing elements, memory storage elements, and communication elements, to produce tiny genetic “ robots” that could reside in cells. This

would allow a level of interface with living processes on a microscopic level that is not possible using strictly silicon-based computing technology.

Such a technique could provide an unprecedented level of control over such processes. This could offer a way for humans to ‘self-treat’ a number of presently untreatable genetically-based illnesses, while allowing a team of tiny “physicians” to remain on call to correct other problems as they arise.

#### 4.4 DNA Data Storage.

The other strong point in favor of biological computing outlined by the experts is the potential

storage capacity of DNA within bacteria [29].

Experiments have successfully retrieved 57-99 base pairs of non-native information from a bacterium.

Considering that a millilitre of liquid can contain  $10^9$  bacteria, if we assume an average of 80 base pairs of information per bacterium, using 4 base pairs per byte that would give 19 Gigabytes of data for a volume of one millilitre. In comparison the current areal density record for magnetic storage stands at 10 Gigabytes per square inch. If we consider that the hard-disk platter is 1cm thick that gives 1.5 Gigabytes per millilitre. Therefore the storage density of DNA within bacteria fares strongly against traditional storage mediums.

As the amount of stored data grows, so does the overhead needed to keep the data in order. For

109 bacteria, 30 bits or 15 base pairs are needed to uniquely identify each bacterium. Even though the rate of growth for the needed sequence number space is  $O(\log)2^n$ , this turns out to be a significant percentage of the current usable capacity.

## 5. Conclusions.

Experimental and theoretical research in alternative computing paradigms is accelerating

world-wide. New technologies for realizing quantum and biological computers are being proposed

[30, 31], and new types of alternative computation with various advantages over classical computation are continually being discovered and analyzed and we believe some of them will bear technological fruit.

## 6. References

[1] Feynman R. P., “ Miniaturization,” Reinhold Publishing Corporation, New York, pp. 282-296, 1961.

[2] Simon Bone, “ An Introduction to Quantum Computers,” 1997.

[3] A. Maurya, A. Nair, S. Sanyal, “ An Overview of The Evolutionary Trends in Molecular Computing using DNA,” Accepted for publication in International Journal Of Computing, Information Technology, And Engineering (IJCITAE), December 2007.



[4] Corne D., and Shapiro J. L., editors, “ Evolutionary Computing,” volume 1305, Lecture Notes in Computer Science, Springer-Verlag, 1997.

[5] G. Paun, G. Rozenberg, A. Salomaa, “ DNA Computing: New Computing Paradigms,” Texts in Theoretical Computer Science, EATC Series, Springer-Verlag, 2006.

[6] Heisenberg, “ Uber den anschaulichen Inhalt der quantentheoretischen Kinematik und

Mechanik”, Zeitschrift fur Physik, 1927, pp. 172-198. English translation: J. A. Wheeler and

H. Zurek, “ Quantum Theory and Measurement,” Princeton University Press, 1983, pp. 62-84.

[7] David Deutsch, “ The Fabric of Reality: The Science of Parallel Universes – and Its

Implications,” The Penguin Press, 1997). 6

[8] David Harel, “ Algorithmics: The Spirit of Computing,” Addison-Wesley, second edition, 1992.

[9] J. E. Mooij, T. P. Orlando, L. Levitov, L. Tian, C. H. Van, “ Josephson Persistent-Current

Qubit-Science,” 1999.

[10] “ Principles of Nuclear Magnetic Resonance in One and Two Dimensions,” 1990.

[11] R. R. Ernst, G. Bodenhausen, A. Wokaun, J. Brown, “ A Quantum Revolution for Computing,” New Scientist, 1994.

[12] David P. DiVincenzo, “ Quantum Computation,” Science Volume 270, 1995.

[13] Ekert, “ Quantum Cryptanalysis-Introduction,” <http://eve.physics.ox.ac.uk/QCresearch/cryptanalysis/qc.html>

[14] Ivars Peterson, “ Quantum – Quick Queries,” Science News Volume 150, 1996.

[15] Wong, Wong, Foote, “ Organic Data Memory –Using the DNA Approach,” Comm. ACM, 2003.

[16] R. J. Lipton, “ Speeding up computations via Molecular Biology,” Princeton University Draft, 1994.

[17] Powledge T., “ The Polymerase Chain Reaction,” <http://www.faseb.org/opa/bloodsupply/pcr.html>

[18] Turing A. M., “ On Computable Numbers, with an Application to the Entscheidungs problem,”

1936.

[19] J. C. Cox, D. S. Cohen, A. D. Ellington, “ Trends in Biotechnology – The complexities of

DNA Computation,” 1999.

[20] Adleman, L. M., “ Molecular Computation of Solutions to Combinatorial Problems,” Science,

266: 1021- 1024, November 11, 1994.

[21] Weiss R. & Basu S., “ The Device Physics of Cellular Logic Gates,” First Workshop on Non-

Silicon Computing, Cambridge, Massachusetts, 54-61, February 2002.

[22] Elowitz M. and Leibler S., “ A Synthetic Oscillatory Network of Transcriptional Regulators,” Nature, 403: 335-338, January 2000.

[23] M. Mitchell, “ An Introduction to Genetic Algorithms,” 1996.

[24] Belew R. K. and Vose M. D., editors, “ Foundations of Genetic Algorithms – 4,” pages 117 – 139, Morgan Kaufmann, 1997.

[25] M. Mitchell, S. Forrest, “ Genetic Algorithms and Artificial Life,” 1994.

[26] Levy S., “ Artificial Life,” ISBN: 0224035991, Cape, 1992.

[27] Schena M., Shalon D., Davis R. W., Brown P. O., “ Quantitative Monitoring of Gene

Expression Patterns with a Complementary DNA Microarray,” *Science*, 270 (5235): 467-70, October 1995.

[28] Gardner T., Cantor R., and Collins J., “ Construction of a Genetic Toggle Switch in *Escherichia Coli*,” *Nature*, 403: 339 – 342, January 2000.

[29] “ Data Stored in Multiplying Bacteria,” *NewScientist*, 2003, <http://www.newscientist.com/news/news.jsp?id=ns99993243>

[30] J. Khodor, D. K. Gifford, “ The Efficiency of Sequence-specific Separation of DNA Mixtures

for Biological Computing,” 3rd Annual DIMACS Workshop on DNA Based Computers.

[31] “ Quantum Code-breaking,” *The Economist*, 30

April 1994.