Alternative Modes of Computation

William Gasarch, Nathan Hayes

January 8, 2020

1 Introduction

Since 1965, the idea that the next generation of computer technology would always be twice as small pushed innovation to its limits. Now, however, it seems that the era of Moore's law is coming to a close. Microchip manufacture is beginning to stumble over the limits of physics, and even Moore himself has declared that his law will be dead by 2025. The industry must seek new horizons of development if it is to maintain its momentum.

It has long been known that electronics aren't the only form of computer hardware. In the past, we've used electronics simply because we were able to make them faster, cheaper, and better than anything else, and so we used them even for tasks they were otherwise unsuited for. Without the buoy of Moore's law to support the explosive growth of these devices, however, there is a perfect opportunity for other paradigms of computation to take the stage.

This paper is a review of these various challengers to the throne of the computing industry. We give brief overviews of their nature, assessments of their advantages and disadvantages, notes on the current state of their research and manufacture, and a set of metrics that may be used to evaluate their usefulness.

2 Neuromorphic

2.1 Overview

Neural networks are a type of data structure used in machine learning, and are designed to mimic (in software) the human brain. Layers of 'neurons' represented by vectors carry data forward across 'synapses' represented by matrices, and parameters are adjusted to minimize error. They have shown remarkable success in their capacity to learn from limited data sets, and are used widely for a range of tasks otherwise intractable for standard algorithms, including image processing, speech recognition, data search algorithms, and more. Unfortunately, modern computer architectures are not well optimized for the matrix computations used for neural network computing.

Enter neuromorphic computing: a paradigm of purpose-built computer hardware designed explicitly to perform neural network computations efficiently. By making use of organization similar to that of the human brain, neuromorphic chips gain significant computing advantages for machine learning. As very specialized machines, they are often designed for more specific types of neural network, such as spiking or convolutional neural networks [24]. Neuromorphic computers also take advantage of electrical components that act like biological neurons, such as memristors.

A memristor is a theoretical electrical component that varies its resistance depending on how much current has flowed through it previously. These components open up new possibilities for computer engineering, and are especially good for neuromorphic computers due to their neuron-like behaviour. A range of memristors exist, including titanium dioxide and carbon nanotube memristors, and they are in use by many neuromorphic machines [29]. There are, however, doubts as to whether a "true memristor" can actually exist, and whether all existing ones are merely very good approximations [35] (although they are good enough that we can't tell the difference).

2.2 Advantages

By virtue of its focused design, neuromorphic computation is by far more efficient than standard computation for machine learning, both in theory and in practice. Neuromorphic chips boast significantly reduced power consumption and power density [51]. More importantly, however, neuromorphic computation is much faster and more space efficient than typical computers—it's even more time and space efficient than the human brain [45]. It is also highly programmable, and approachable for use by non-experts.

2.3 Disadvantages

This all comes with some notable caveats. Neural networks as an architecture, for example, are fundamentally unpredictable, in that different trainings can result in different networks with differing levels of success. In addition, neuromorphic computing is fundamentally inefficient for simple mathematical computation, since it has to learn from scratch the rules of arithmetic [2]. Neuromorphic microprocessors also have hit several stumbling blocks when it comes to ease of use, integration with standard computers, and even integration with other neuromorphic processors [25]. Notably, as with all forms of dedicated hardware, neuromorphic architectures may 'phase out' when better versions are developed, since it becomes difficult to run modern software on outdated single-purpose hardware.

2.4 Current Work

Much research is being done on neuromorphic computation, and there are several notable working chips and microprocessors that are designed for this form of computation. The Intel Loihi research chip shows remarkable success, both in terms of speed and energy efficiency, and is designed for asynchronous spiking neural network computation [6]. More recently, IBM's TrueNorth microprocessor has shown similar results [25], but with more success at overcoming integration problems, and is currently in use by DARPA's SyNAPSE processor [34].

2.5 Metrics

Neuromorphic computing is a very specialized category designed explicitly for machine learning. Therefore, the metrics that should be used to evaluate neuromorphic computing are the same ones used to evaluate machine learning architectures, such as time to convergence and training error. In addition, there are hardware considerations: energy efficiency, space efficiency, durability, and integrability with other neuromorphic/classical architectures.

3 Neurological

3.1 Overview

Neurological computation refers to the way biological neurons self-organize to process information. While we know with certainty that it is very effective, scientists have only a limited understanding of the biochemical processes that occur to make the brain function. Sometimes referred to as wetware, as opposed to hardware or software, this form of computation is especially effective when it comes to association and learning processes.

3.2 Advantages

Neurological computing has all the potential advantages of human brains over computers — brains are exceptional at self-organizing and rewriting, at classification, and at learning tasks. In addition, the biochemical nature of neurological systems means they are remarkably energy efficient when compared with classical computers [26].

3.3 Disadvantages

Of course, this also means that neurological computers have all the flaws of human brains. They require delicate physical, biochemical, and thermal conditions, and we lack much of the requisite knowledge to maintain them outside of living organisms. They tend to be computationally disorganized and inefficient for standard tasks, and since they are fundamentally association machines, they have a tendency to make mistakes.

3.4 Current Work

Of course, there are plenty of examples of functioning neurological computers - there's one in every head, human, animal, and even insect. More importantly, we've made headway into understanding them, as the fields of cognitive science and neuroscience clearly demonstrate. There is even an example of a man-made (or rather, man-designed) neurological computer. In 1999, William Ditto created a computer using leech neurons that could perform simple addition [28]. However, it is fair to say we understand so little of the human brain that any form of useful neurological computer is far away, even for research.

3.5 Metrics

Neurological computation is specialized in the same way as neuromorphic computation, and thus should be evaluated by the same metrics - time to convergence, energy consumption, space efficiency, and durability. Notably, however, one should also consider interpretability for this form of computation, as it is not always easy to extract answers. Of course, the lack of understanding of neurological systems presents an obstacle to identifying effective metrics, so this list should be considered incomplete.

4 Spintronics

4.1 Overview

Fundamental particles have a property known as spin, which influences how they interact with magnetic fields. Notably, the measured spin of an electron is always one of two directions: up or down. Spintronics deals with harnessing this property of electron spin for use in solid-state (no moving parts) systems. By deliberately altering and measuring the spins of trapped electrons, we can build spintronic chips and, hopefully, computers. These would function logically in the same way as classical computers, but with multiple practical benefits. This is already being done for information storage [9], but it has yet to be done for computing.

4.2 Advantages

Spintronics has great potential for use in the future as a replacement for, or supplement to, classical computing via storage and processing improvements. We explore both these themes.

Spintronic storage has various hardware advantages, including increased writing speed, space efficiency, resistance to heat and electromagnetic interference, and critically, non-volatility (storage does not require constant power) [12]. These improvements have been demonstrated by existing storage chips, as will be seen later.

Spintronic computation shows great potential for future development in a variety of areas that have yet to be practically realized. These include computation using spintronic transistors, spinplasmonics (see next paragraph) and even integrations of spintronics with quantum and/or neuromorphic computing. It is expected that improvements in these areas will share the speed, space efficiency, and resilience against heat and electromagnetism of existing spintronic storage.

Spinplasmonics is a combination of spintronics with plasmonics, a field studying plasmons. Plasmons are quanta of plasma oscillation, and play a role in the light waves used to read and write to spintronic data. Through use of plasmonics, it is possible to read and write to spintronic data without electronic components, which may accelerate read and write speeds [22]. The major hope, however, is that spinplasmonics may enable further miniaturization of spintronic computing.

The idea of integrating spintronics with quantum computing is founded on the observation that a particle's spin can behave as a qubit. Then, by combining spintronics' ability to manipulate spin with quantum's ability to compute with it, we may improve the efficiency of quantum computation. This goal rests far in the future, however, as quantum computation is still a growing field, and spintronic techniques lack the power as of yet to accomplish this integration. Neuromorphic computing, on the other hand, can make use of spintronics in a simpler way, the same way as classical computation does. This entails using efficient spintronic processing and memory for neuromorphic computation. Thus, integrations of neuromorphic and spintronic computing are more promising for the near future, since the technology to integrate is better-understood.

4.3 Disadvantages

There are issues when integrating spintronics with classical computing. In particular, cross-platform information transfer is energy costly and slow. There are also trade-offs for various hardware constraints of spintronic storage: good retention means high power consumption, and low power consumption degrades retention, for example [50]. At the moment, spintronic MRAM (magnetic random access memory) is very space- and cost-inefficient [3], though further research may alleviate this. In addition, although this technology shows great promise, spintronic processing does not show any signs of reducing the number of steps of any computation (although it does reduce the time it takes to do one step); in short, it lacks a "killer application," such as quantum computing's potential to lower the time complexity of factoring from exponential to polynomial time.

4.4 Current Work

Many microchip and electronics companies are investing a great deal of time and money into spintronics research. Some of them have developed, tested, and even sold working MRAM chips used for data storage that use spintronic techniques. Notably, Everspin has developed working STT-MRAM (Spin-transfer torque) chips that have been incorporated into working demonstration computers by Intel, Samsung, and others [20]. IBM, too, is breaking ground with experimental but incredibly efficient Racetrack Memory chips that may potentially outperform almost all modern computer memory chips [4]. Spintronic processors, however, are yet to be practically realized, though current research suggests they may be in the near future.

4.5 Metrics

Spintronic computation is not specialized as of now - it computes in the same way as classical computers, but with different fundamental components. For this reason, the valuable metrics for spintronics are hardware metrics, since having hardware advantages over electronic computers is what makes it viable as a form of alternative computation. The hardware metrics worth considering include space efficiency, speed, energy efficiency, and durability. Also notable as a useful way of tracking progress in spintronic research is the accessibility of spintronic devices for non-expert use.

5 Quantum

5.1 Overview

Classical computers store and manipulate information as bits, which can each be in one of two discrete states (1 and 0). Quantum computers, however, manipulate qubits, special structures that utilize the quantum properties of matter and can exist in the states of 0, 1, or any superposition of the two. However, when measured, qubits "collapse" to either 1 or 0. By using complex superpositions of qubits, quantum computers can compute things quickly (and often probabilistically) that classical computers cannot.

5.2 Advantages

Essentially, quantum computing is a form of powerful parallel processing with strong limitations on its readout; although a set of n qubits can store n complex numbers worth of information, only n bits of it can be accessed. The freedom to manipulate which bits are accessed is what gives quantum computation its power, but figuring out how to do so is difficult. The Deutsch problem serves as a simple example.

Suppose you have a function f:$\{0,1\} \rightarrow \{0,1\}$. Clearly, there are only 4 such functions : f(x) = x, f(x) = 1-x, f(x) = 0, and f(x) = 1. We can say that f(x) = x and f(x) = 1-x are balanced and f(x) = 0 and f(x) = 1 are constant. Suppose we are given a black box that computes one of these functions, and we want to determine if it is balanced or constant in as few evaluations as possible. From an information theory point of view, we only desire one bit of information, and each function evaluation gives us one bit, so we'd like it to be possible in one evaluation. With a classical computer, of course, this is impossible - we must evaluate the function twice. A quantum computer, however, can determine if the function is balanced or constant in a single evaluation, by performing the function on a superposition of 0 and 1, and then changing the bases of measurement such that balanced functions evaluate to 1 and constant functions to 0.

This technique can be used for a variety of quantum algorithms. The most famous are Shor's and Grover's algorithms for discrete log and unsorted search, but there are a variety of lesser-known algorithms that have been developed. Many are algebraic and number theoretic in nature, like Shor's algorithm, and some are "oracular," like Grover's, but a number are simulation algorithms [30].

Quantum is probably best known for its applications to number theory, and especially for Shor's algorithms for factoring and discrete log. Constraint satisfaction problems, subset sum, matrix product verification, and various abstract algebra problems are possible with quantum computing [30], and some algorithms for these problems have superpolynomial improvements over their classical counterparts, which is the reason for the majority of the buzz surrounding quantum.

Oracular algorithms deal with "oracle problems," or problems in which we desire some information about a difficult-to-analyze function. The Deutsch problem seen above is one of the simplest examples of this, and has been extended into the Deutsch-Josza algorithm for n-bit cases of the problem. These algorithms cover a wide range of topics [30], including boolean evaluation, linear systems, graph properties, and even machine learning [30] [49].

When quantum computing was first introduced, many of its proponents, including Richard Feynman, were most excited about its potential to simulate quantum systems [52]. This remains a major motivation for quantum research, and other forms of simulation and approximation algorithms have been devised that show potential. These algorithms cover a range of topics including knot theory, manifolds, and linear algebra [30].

5.3 Disadvantages

Quantum computation has its flaws, however. As of now, no quantum computer has been able to outpace classical computers at factoring, although they continue to shrink the gap. Beyond this, however, there are more fundamental issues. Quantum computers are very informationally noisy [43], so there are limits on the accuracy of computation. In addition, most quantum computers must be supercooled to function. Even if all of this is addressed, however, quantum computing only has a small set of problems it is expected to be able to solve. Finally, quantum machines require high levels of expertise to use, and programming them efficiently is still a growing and very technically difficult field.

5.4 Current Work

Quantum technology has made great strides recently. Prior to 2016, the most qubits any quantum machine has been able to use was 12, but since then there has been an explosion of corporate interest. Many companies now have working quantum computers and chips used for research purposes, such as Google's 72 qubit Bristlecone [32], Intel's 49 qubit Tangle Lake [5], and IBM's unnamed 50 qubit prototype [33]. IBM has even released a "commercial" (being sold to a limited group for research) quantum computer, the Q System One [7], a 9-foot cube with 20 qubits, and is selling computation time on it to other companies and labs. It must be noted that there is a company called D-Wave that claims to have surpassed 2000 qubits with a technique called "quantum annealing," but there are heavy doubts as to whether their computers classify as truly quantum [14]. There is much more work to be done, however, as all of these processors are designed for research, rather than practice.

5.5 Metrics

Quantum computing is a very atypical form of computing, as it is not used for routine computation, but rather for remarkably specific applications that are still in progress. In addition, many major quantum algorithms, such as Shor's and Grover's algorithms, are probabilistic. As such, the most valuable metrics to use are unusual, such as the largest number it has factored, the temperature required, and the probability of algorithm success. Some, however, are more typical, such as the time complexity of algorithms for quantum computation.

6 Optical

6.1 Overview

There are several potential paradigms of optical computation, since it belongs in a mostly theoretical domain. The simplest of these paradigms is the same as typical computers with all electrical components replaced or supplemented by optical components. Some other paradigms, however, are less straightforward. For example, time delay optical computing involves splitting and delaying beams of light to perform highly parallel processes.

6.2 Advantages

All forms of optical computing have an edge over classical computers in heat, resistance to interference, and energy consumption [8]. This is because electrons interact more directly with matter than photons, which wastes energy and makes them susceptible to the environment. In addition, optical computing shows promise for image processing and machine learning. Some theoretical designs even show potential in solving NP-Hard problems such as the subset-sum problem [46].

6.3 Disadvantages

Unfortunately, optical computers have difficulty interfacing with classical computers due to constraints on the efficiency of converting information between light and electricity [38]. Also notably, the theoretical designs that promise computational complexity improvements also promise abysmal performance on standard problems. When coupled with integration problems, this poses a challenge to complex problem solving for any general-purpose optical computer.

6.4 Current Work

While there are a number of theoretical designs for optical computing devices, very few actually exist. Some, however, have made it to the point of commercial availability - Optalysys launched the first commercially available optical processing system, the FT:X 2000, on March 7th of 2019 [8]. The FT:X is an AI co-processor designed for image and video machine learning applications. Beyond that, however, nothing of much impact has left the research phase.

6.5 Metrics

The metrics that are useful to evaluating optical computing vary depending on the optical architecture being examined. For classical computers using optical components, the metrics worth considering are hardware metrics, such as speed, heat, energy efficiency, and durability. For other, more specialized versions, they must be evaluated on the metrics relating to the problems they solve. For example, the FT:X should be evaluated on time to convergence for visual data processing, etc. Work with optical processors requires some level of familiarity, however.

7 Chemical

7.1 Overview

Chemical computation refers to computation using chemical reactions, and is relatively broad in that there are several theoretical models of how one might function. The most studied example is the Belousov-Zhabotinsky (BZ) computer, also known as the reaction-diffusion computer. These computers use the spread of a periodic reaction and the interaction between concentration waves to compute. The reaction can be modified to be light-sensitive, and so is researched in connection with image processing [44]. Another, less developed paradigm is molecular computing, which uses individual molecules as data and logical gates. Some aspects of molecular computing show intersection with amorphous computing, which will be covered in the next section.

7.2 Advantages

Chemical computing is fundamentally completely different from classical computing, both in processing and in information storage. For clarity, we examine both these concepts separately.

All forms of chemical computer show potential to be excellent parallel processing tools, due to the non-sequential nature of chemical reactions. BZ computers show potential to be used as effective image-processing tools due to their light-sensitive variants. Molecular computing also shows great parallel processing potential, perhaps even more that BZ.

BZ computers don't show much potential for storage improvements, but molecular computers do to an immense degree. By storing information in molecular arrangements, molecular computers could surpass the Moore's Law limit and potentially store information in as little space as is physically possible under any scheme.

7.3 Disadvantages

Individual computational steps on chemical computers take a long time (in contrast with, say, spintronics, where individual steps are faster than classical computers). Chemical reactions are fundamentally slow, so when it isn't possible to take advantage of parallel processing chemical computers can't keep pace. Extracting answers from chemical computers can also be difficult, since measuring results visually requires the addition of indicator chemicals that may be slow or unreliable [21]. The major hurdle, however, is the difficulty of programming chemical computers with useful or lengthy instructions, as the molecules required to do this may be unstable or hard to synthesize. In addition, it requires expertise both in computer science and in chemistry to effectively operate chemical computers, and it is difficult to predict whether this will change in the future.

BZ computers show no real storage improvements at all, and in fact may be less information dense than classical computers. Read and write times for BZ computers are also not promising, though reading and writing may be conducted in parallel. Molecular computers also present challenges, as most likely storage will be difficult to access, and may be unordered.

7.4 Current Work

Several proof-of-concept BZ-computers have been developed, mostly for image processing. One such computer was used to compute the shortest path between two points in a maze without having to check every path, like a GPS-style computation would [48]. Research is being done to improve upon these models and design them to perform more useful computation. In addition, numerous computing molecules have been designed and tested [42], though none yet demonstrate promising complexity of computation.

7.5 Metrics

Chemical processing is strange in that it functions in a fundamentally stochastic, non-sequential fashion. As such, it is necessary to compare it both on linear and parallel terms. For this reason, the metrics that should be examined include speed of linear computation and speed of parallel computation. In addition, specialized chemical computers for tasks such as image processing should also be evaluated on specific criteria pertaining to those tasks, such as reliability and resolution.

For chemical storage, other areas require quantification. Since BZ computers will likely lower data storage density, while molecular computers will increase it, both should be evaluated on that metric. Reading and writing times, too, should be considered, as accessing data will be difficult. In fact, both paradigms struggle with the fundamental issue of interpretability of results, and so this metric should be addressed as concretely as possible.

8 Amorphous

8.1 Overview

The term "amorphous computing" was coined in the Amorphous Computing Manifesto [10] in 1996. Amorphous computation, in the broadest sense, is computing with emergent properties of self-organizing systems. More specifically, an amorphous computer is a interconnected network of simple processors, each with no prior knowledge of its position or anything beyond its neighborhood, operating asynchronously (without a shared clock). This paradigm is notable in that it is hardware-agnostic; an amorphous computer could be a collection of inter-communicative robots, a cluster of cells, or even a well-designed chemical solution.

8.2 Advantages

Of course, the most major benefit of this style of computation is parallelism, the natural advantage of aggregating many small processors together. Further, however, amorphous computation is an excellent model for many forms of natural system. Biological cells operate together as amorphous computers; for example, the cells of a leopard's skin communicate via chemical signals to form a distinctive rosette pattern [37]. Similar mechanisms regulate the heartbeats and brainwaves of living organisms.

Amorphous computing also shows potential to be used for specific forms of microfabrication, using a thin sheet of microprocessors that can change state to self-organize into electronic circuits [19]. Similarly, proponents of the paradigm hope that its potential to organize complex behavior from simple parts may fuel a new generation of smart materials. On a more macro scale, MIT labs have already begun testing amorphous computing concepts and algorithms with ant-like robot networks and self-assembling structures [13].

8.3 Disadvantages

By nature, amorphous computers are random, inexact, and do not preserve information well. They are not well suited for any form of sequential operation, and the design constraints of cells with no prior knowledge of their locations or surroundings severely cripple even simple tasks. Addressing all cells of the computer, for example, requires random addressing over a large enough space to preclude duplicates [10], which means that cells must have prior knowledge of the size of the computer, and even this may not be desired. They are poorly understood, and while they are programmable, we don't understand well how to design effective algorithms for amorphous computers.

The largest disadvantage to amorphous computation, however, is the fact that any centrally-organized classical computer of the same size as an amorphous one is strictly better for all sequential operations. Classical computers are even usually better for parallel computation due to the difficulty of designing amorphous algorithms that can effectively parallelize computation. For this reason, amorphous computers are only more effective computation tools for very specific problems.

8.4 Current Work

The principles of amorphous computing are plainly visible in the natural world, especially in cellular biology. Human-made amorphous computers, however, largely remain in the theoretical realm. That said, there is a growing body of research on amorphous computing, including investigation of error correction [18], self-repairing systems [53], and amorphous computing languages such as the Growing Point Language [19]. Theoretical models of amorphous computers show promise for certain parallel tasks, as do real-world biological amorphous computers like Physarum Polycephalum, a yellow slime mold with remarkable intelligence. That said, given how new the field is, it's very difficult to tell how research will develop.

Other approaches involve using chemical processes to create chemical simulated annealing machines based on the Ising model. This claims to have the advantage being able to approximate solutions to NP-hard problems like the satisfiability problem.

8.5 Metrics

Amorphous computing is a very young field, even relative to the other forms of largely theoretical computation described herein. For this reason, all of the following examples of metrics that could be used to evaluate an amorphous computer should be taken with a grain of salt.

Amorphous computing, like chemical computing, is fundamentally asynchronous and stochastic. As such, it is necessary to compare it both on linear and parallel terms. For this reason, the speed of both linear and parallel computation should be examined. In fact, given the (currently) small space of problems for which amorphous computing provides any speed advantage, the size of said space can be considered an important metric of evaluation.

Given amorphous computing's unique propensity for microfabrication, various metrics relating to self-assembling materials should also be considered. These, of course, include the size and speed of the microprocessors used, as well as their physical integrity for certain applications. In addition, various considerations including communication range and reliability, microprocessor failure rate, and even resistance to malicious interference should be considerations.

9 DNA

9.1 Overview

DNA is a chemical chain consisting of 4 chemical bases, and it is stored in the cells of a wide variety of organisms. It stores a massive amount of information and is used as the blueprint for all of the proteins the body produces. The intent of DNA computing is to harness the preexisting chemical structure and properties of DNA to store and manipulate information encoded into those 4 chemical bases.

9.2 Advantages

We know that DNA computing has potential, because our bodies compute with DNA every minute of every day, replicating it and using it to grow new cells. Since it is a molecular-level structure, DNA is incredibly space efficient [47], and moreover is very well suited for massive parallelism. This means that DNA computing has the potential to reach incredibly high speed for problems that can be effectively parallelized. Thus, it is claimed that DNA computers might be able to solve problems such as evaluating boolean circuits [39].

9.3 Disadvantages

Unfortunately, DNA computing is limited by the speed of the reactions with which it can be edited, copied, and read. As such, even though it is massively parallel, it is limited in speed by the inefficiency of basic computation. In addition, it is a tremendous challenge to read and interpret data produced by a DNA computer [16], since it is such a microscale process. Finally, it requires a relatively large expenditure of chemical resources to program and run an algorithm on a DNA computer.

9.4 Current Work

An impressive amount of success has been attained by the field of DNA computing research. DNA computing has been used to solve the traveling salesman problem on 7 cities [11] and 3-SAT on 20 variables [15], and even to play Tic-Tac-Toe [23]. Notably, however, is the fact that although it has solved these problems, in many cases the answers could not be extracted, because it is difficult to interpret answers from DNA computation. In addition, the TSP and 3-SAT results are from 1994 and 2002 respectively, and while work has built on this research, no breakthroughs have granted DNA computing practical viability. Still, research into DNA computing shows impressive potential.

Current research has also suggested ways to overcome poor DNA synthesis speed - a team at the University of Illinois has developed a scheme that involves ”nicking” native E. coli DNA to store information [36], effectively negating the cost of synthesis. This scheme also has a highly parallelizable theoretical language to allow for in-memory computation, limiting readout costs. This is still deeply in the theoretical realm, however, and has major logistical concerns relating to computation fidelity. This has led to the consideration of employing stochastic computing methods. For example, one could randomly nick a strand of DNA with a frequency of 1/n and another strand with 1/m. By taking the logical AND of these strands, one could approximate 1/mn, allowing for simple multiplicative computation.

9.5 Metrics

The major advantages and disadvantages of DNA computing lie in its chemical nature, which leads it to be highly parallel and information-dense. The metrics best for evaluating this are speed of sequential computation, speed of parallel computation, data storage density and longevity, and interpretability of solutions. While the final metric is difficult to evaluate, it is critical to consider when examining the viability of DNA computing due the importance that must be given to resolving this problem.

10 Peptide

10.1 Overview

Peptides are chemical chains consisting of 20 types of amino acid, and they are used for many different biological processes. They, like DNA, can store much information in the sequence of amino acids they consist of. The intent of peptide computing is to harness the information storage and manipulation capabilities of peptides to perform computation.

10.2 Advantages

Peptide computing is very similar to DNA computing, and thus has many of the same advantages, primarily extremely space efficient data storage and massive parallelism. It also has notable advantages over DNA in a few ways, however. First, with 20 amino acids instead of 4 bases, peptide computing has the potential to be even more space efficient than DNA. Second, peptide reactions are more flexible and can more easily accept antibodies for reaction [27].

10.3 Disadvantages

Of course, peptide computing has all of the downsides of DNA computing as well. Not only does it present no potential for faster simple computation than DNA, it also shows no signs of being any easier to extract results from. The major roadblock, however, is the lack of availability of some monoclonal antibodies which would be necessary to realize a working peptide computer [27]. The details of what monoclonal antibodies are and why they are necessary do not concern us here, except to note that their scarcity presents an obstacle for peptide computing.

10.4 Current Work

Although no working peptide computers exist, there is much substantive research on how one might be practically realized. A paper by Hug and Schuler [27] describes how a peptide computer might be used to solve NP-Complete problems like the satisfiability problem. The pace of research remains steady, despite the lack of experimental results.

10.5 Metrics

Peptide computing is incredibly similar to DNA computing. The major advantages and disadvantages of peptide computing lie in its chemical nature, which leads it to be highly parallel and information-dense. The metrics best for evaluating this are speed of sequential computation, speed of parallel computation, data storage density, and interpretability of solutions. While the final metric is difficult to evaluate, it is critical to consider when examining the viability of peptide computing due to the importance that must be given to resolving this problem.

11 Membrane

11.1 Overview

Membrane computation is a theoretical computational model inspired by the function of cellular membranes. It studies nested ”membranes” in various arrangements, known as P-systems, and the evolution rules they follow that lead to desirable computations. This model of computation is not a physical system, but rather a theoretical area of study, and has no direct hardware analog from which it is derived. The closest version of such a thing is the behavior of cell membranes, but the theoretical system and the physical system are fundamentally different in nature.

11.2 Advantages

P-systems have the advantage of being somewhat efficient for computing in parallel, and describe effectively the properties of self-organizing systems [40]. In addition, since similar systems are in play within our bodies, membrane computing could be used to understand bodily computing and vice versa.

11.3 Disadvantages

Membrane computing is an entirely theoretical computing paradigm, offering no major practical benefits and with no working hardware models. It is unknown how to implement most necessary and useful algorithms using membrane computation, even including simple programming concepts such as recursion.

11.4 Current Work

There exists an abandoned patent for a potential liquid model of a membrane computer [31]. Otherwise, all work done is theoretical and still in progress. Though a substantial amount of theoretical work has been done, no satisfactory physically implementable systems yet exist.

11.5 Metrics

Seeing as no practical models of membrane computing exist, viable metrics are hard to identify. However, since it is designed as an alternative to classical computing that is similarly universal and is focused on parallel processing, we can extrapolate from chemical computing what may be necessary. The metrics that should be examined include the time complexity of linear computation and the time complexity of parallel computation. Of course, the lack of existing systems to draw conclusions from means this list is woefully incomplete.

12 Fluidic

12.1 Overview

Fluidic computation uses and collides jets of water to create logic gates. These logic gates are aggregated into a physical system, much like a microchip, to perform computations. For example, one can simulate an OR gate with two jets that meet into a stream. When either jet is on, water flows through the stream and the gate is active.

12.2 Advantages

Fluidic computing is most applicable in areas of nanotechnology, where miniaturization is paramount and so classical computing is difficult or impossible. There, microfluidic systems may be able to be integrated into nanoscale machines for processing. Microfluidic systems implementing bubble logic [41] can carry chemical payloads, allowing for both computation and material manipulation simultaneously. Microfluidic systems can also be used as effective simulations for biological systems, or even as time-delay drug delivery mechanisms, and are therefore useful for medical applications [1]. While it may not be immediately obvious that this can be considered computing, we take an expansive view, and therefore argue that it qualifies. Certainly, whether it qualifies or not, it would be useful if fully developed. For more large-scale fluidic systems, the primary advantage is hardware flexibility.

12.3 Disadvantages

Fluidic computation is incredibly difficult to reprogram, since there are no moving parts, and so most fluidic systems are limited to predesigned single-use systems. Fluidic systems are both space-, time-, and energy-inefficient, and heat sensitive. They have no theoretical computation benefits when not used as analog systems.

12.4 Current Work

Given that no papers appear when searching Google Scholar for ”fluidic computer,” it is safe to say that no serious research is being done into the subject. Several amateur enthusiasts have designed and built such computers, as have some students at Stanford [17], but these are entertainment projects rather than practical designs. For biomedical applications, however, tangent work is being done into miniature, microfluidic systems for injection and physiological simulation [1].

12.5 Metrics

The valuable metrics for fluidics are hardware metrics, such as space efficiency, speed, durability, and bio-compatibility. This is because fluidic computation is not specialized, and is primarily useful for non-programmable (much like a circuit), bio-compatible applications.

References

[1] Microfluidics as a tool for drug delivery. https://www.elveflow.com/microfluidic-tutorials/cell-biology-imaging-reviews-and-tutorials/microfluidic-for-cell-biology/microfluidics-as-a-tool-for-drug-delivery/.

[2] Neuromorphichips Pros and Cons. https://sites.google.com/site/neuromorphichips/pros-and-cons.

[3] Informationstoragecourse2007. http://bluwiki.com/go/Informationstoragecourse2007, 2007.

[4] Memory with a spin. *Nature Nanotechnology*, 10:185, Mar 2015.

[5] 2018 ces: Intel advances quantum and neuromorphic computing research. https://newsroom.intel.com/news/intel-advances-quantum-neuromorphic-computing-research/gs.knxkhx, Jan 2018.

[6] Intel announces neuromorphic research progress. https://newsroom.intel.com/news/intel-announces-neuromorphic-computing-research-collaborators/gs.k0elhs, Dec 2018.

[7] Ibm unveils world's first integrated quantum computing system for commercial use. https://newsroom.ibm.com/2019-01-08-IBM-Unveils-Worlds-First-Integrated-Quantum-Computing-System-for-Commercial-Use, Jan 2019.

[8] Optalysys launches FT:X 2000 The worlds first commercial optical processing system. https://insidehpc.com/2019/03/optalysys-launches-ftx-2000-the-worlds-first-commercial-optical-processing-system/, Mar 2019.

[9] STT-MRAM: Introduction and market status. https://www.mram-info.com/stt-mram, Feb 2019.

[10] H. Abelson, T. F. Knight, G. J. Sussman, et al. Amorphous computing manifesto. https://groups.csail.mit.edu/mac/projects/amorphous/white-paper/amorph-new/amorph-new.html, June 1996.

[11] L. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994.

[12] M. D. Amipara. Nano technology - spintronics. *IOSR-Journal of Electronics and Communication Engineering*, pages 14–18, 2013. http://www.iosrjournals.org/iosr-jece/papers/sicete-volume3/30.pdf.

[13] J. Bachrach, J. McLurkin, and A. Grue. Protoswarm: A language for programming multi-robot systems using the amorphous medium abstraction. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '08, pages 1175–1178, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[14] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer. Evidence for quantum annealing with more than one hundred qubits. *Nature Physics*, 10(3):218–224, Mar 2014.

[15] R. S. Braich, N. Chelyapov, C. Johnson, P. W. K. Rothemund, and L. Adleman. Solution of a 20-variable 3-sat problem on a dna computer. *Science*, 296(5567):499–502, 2002.

[16] R. S. Braich, C. Johnson, P. W. K. Rothemund, D. Hwang, N. Chelyapov, and L. M. Adleman. *Solution of a satisfiability problem on a gel-based DNA computer*, pages 27–42. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[17] B. Carey. Just add water: Stanford engineers develop a computer that operates on water droplets, June 2015.

[18] D. Chu, D. J. Barnes, and S. Perkins. Amorphous computing in the presence of stochastic disturbances. *Biosystems*, 125:32 – 42, 2014.

[19] D. Coore. Introduction to amorphous computing. In *Proceedings of the 2004 International Conference on Unconventional Programming Paradigms*, UPP'04, pages 99–109, Berlin, Heidelberg, 2005. Springer-Verlag.

[20] T. Coughlin. MRAM Developer Day. https://www.forbes.com/sites/tomcoughlin/2018/08/10/mram-developer-day/370ddde56c6d, Aug 2018.

[21] L. Cronin, J. M. P. Gutirrez, S. Tsuda, A. Sharma, G. Cooper, G. Aragon-Camarasa, and K. Donkers. A programmable chemical computer with memory and pattern recognition. 2 2019.

[22] A. Y. Elezzabi. Spinplasmonics: a new route for active plasmonics. 6641, 2007.

[23] M. Elstner and A. Schiller. Playing tic-tac-toe with a sugar-based molecular computer. *Journal of Chemical Information and Modeling*, 55(8):1547–1551, 2015. PMID: 26203667.

[24] N. Hemsoth. First wave of spiking neural network hardware hit. https://www.nextplatform.com/2018/09/11/first-wave-of-spiking-neural-network-hardware-hits/, Sept 2018.

[25] N. Hemsoth. A Rare Peek into IBMs True North Neuromorphic Chip. https://www.nextplatform.com/2018/09/27/a-rare-peek-into-ibms-true-north-neuromorphic-chip/, Sept 2018.

[26] J. Hruska. The human brains remarkably low power consumption, and how computers might mimic its efficiency. https://www.extremetech.com/extreme/185984-the-human-brains-remarkably-low-power-consumption-and-how-computers-might-mimic-its-efficiency, July 2014.

[27] H. Hug and R. Schuler. Strategies for the development of a peptide computer . *Bioinformatics*, 17(4):364–368, 04 2001.

[28] R. C. Johnson. Leech neurons enter computer slug-fest. https://www.eetimes.com/document.asp?doc$_id = 1139800, July 1999$.

[29] R. C. Johnson. Startup Beats HP, Hynix to Memristor Learning. https://www.eetimes.com/document.asp?doc$_id = 1327068, July 2015$.

[30] S. Jordan. Quantum algorithm zoo. quantumalgorithmzoo.org/, June 2018.

[31] E. Keinan. Membrane computing, May 2009.

[32] J. Kelly. A preview of bristlecone, googles new quantum processor. https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html, Mar 2018.

[33] W. Knight. Ibm raises the bar with a 50-qubit quantum computer. https://www.technologyreview.com/s/609451/ibm-raises-the-bar-with-a-50-qubit-quantum-computer/, Nov 2017.

[34] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[35] P. Meuffels and R. Soni. Fundamental issues and problems in the realization of memristors. July 2012. https://arxiv.org/abs/1207.7319.

[36] O. Milenkovic. Dna-based data storage and computing. http://shannon.engr.tamu.edu/wp-content/uploads/sites/138/2018/05/Texas-Part2.pdf, 5 2018.

[37] J. D. Murray. How the leopard gets its spots. *Scientific American*, 258(3):80–87, 1988.

[38] D. D. Nolte. *Mind at Light Speed: A New Kind of Intelligence*, page 34. The Free Press, 2001.

[39] M. Ogihara and A. Ray. Simulating boolean circuits on a dna computer. *Algorithmica*, 25(2):239–250, Jun 1999.

[40] G. Paun and G. Rozenberg. A guide to membrane computing. *Theor. Comput. Sci.*, 287(1):73–100, Sept. 2002.

[41] M. Prakash and N. Gershenfeld. Microfluidic bubble logic. *Science*, 315(5813):832–835, 2007.

[42] A. Prasanna de Silva and N. D. McClenaghan. Proof-of-principle of molecular-scale arithmetic. *Journal of the American Chemical Society*, 122(16):3965–3966, 2000.

[43] J. Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, Aug. 2018.

[44] N. G. Rambidi, K. E. Shamayaev, and G. Y. Peshkov. Image processing using light-sensitive chemical waves. *Physics Letters A*, 298(5-6):375–382, Jun 2002.

[45] S. Reardon. Artificial neurons compute faster than the human brain. https://www.nature.com/articles/d41586-018-01290-0, Jan 2018.

[46] N. T. Shaked, S. Messika, S. Dolev, and J. Rosen. Optical solution for bounded NP-complete problems. , 46(5):711–724, Feb 2007.

[47] S. L. Shipman, J. Nivala, J. D. Macklis, and G. M. Church. CRISPRCas encoding of a digital movie into the genomes of a population of living bacteria. *Nature*, 547:345–349, July 2017. https://www.nature.com/articles/nature23017supplementary-information.

[48] D. Szondy. Empa invents chemical computer faster than a satnav. https://newatlas.com/chemical-gps/34446/, Oct 2014.

[49] N. Wiebe, A. Kapoor, and K. M. Svore. Quantum Perceptron Models. *arXiv e-prints*, page arXiv:1602.04799, Feb 2016.

[50] W. G. Wong. Unleashing mram as persistent memory, Mar 2018.

[51] T. Wunderlich, A. F. Kungl, E. Mller, A. Hartel, Y. Stradmann, S. A. Aamir, A. Grbl, A. Heimbrecht, K. Schreiber, D. Stckel, C. Pehle, S. Billaudelle, G. Kiene, C. Mauch, J. Schemmel, K. Meier, and M. A. Petrovici. Demonstrating Advantages of Neuromorphic Computation: A Pilot Study. *Frontiers in Neuromorphic Engineering*, Nov 2018. https://doi.org/10.3389/fnins.2019.00260.

[52] C. Zalka. Simulating Quantum Systems on a Quantum Computer. *Proceedings: Mathematical, Physical and Engineering Sciences Proceedings: Mathematical, Physical and Engineering Sciences*, 454:313–322, Jan 1998.

[53] J. Zucker. Self-healing stuctures in amorphous computing. June 2000.