Chapter 2

Literature Review

DNA computing is a subsection of molecular computing. In this unconventional computational technique DNA molecules are employed as computational unit instead of traditional silicon based technology because of its massive parallel information processing property and compact storage capacity. In DNA computing inputs, processing unit as well as the outputs are in the form of encoded strands of DNA or other physical or chemical agents that manipulates DNA strand. Though initially DNA computing research was confined within the context of solving Non-deterministic Polynomial class problems (NP), but eventually it spreads to other areas as well.

This chapter dedicated to a literature survey covering most of the honorable works in the field of DNA computing with special emphasis on several DNA based logic gate and Boolean circuit simulation models. The chapter ends with critical analysis of some of the DNA computing models in **Section 2.1** and **Section 2.2**.

2.1 An Overview on DNA Computing Models

Fynman realized the importance of interdisciplinary research as early as in 1960 and published the scope on molecular level miniaturization [34]. L. Adleman [1, 2] a computer scientist for the first time demonstrated the computational property of DNA molecules in 1994 when he successfully experimentally validated his idea of solving a small instance of a directed Hamiltonian Path Problem (HPP). HPP is an NP class problem where the task is to find a path between the start and end node in a graph G with n vertices visiting every node exactly once. The graph solved by Adleman using DNA computing approach is shown in **Figure 2.1**.

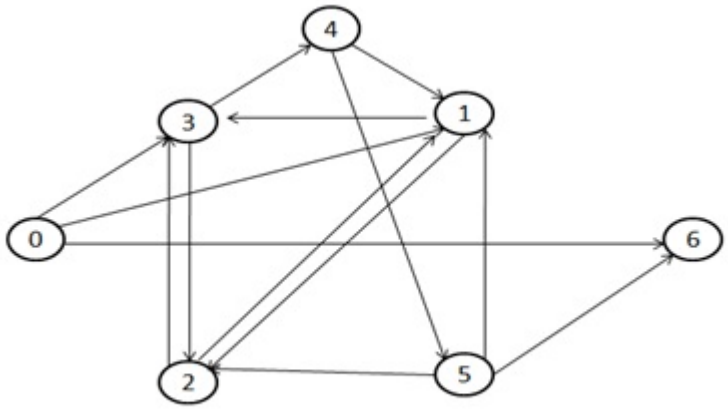

Figure 2.1: Directed graph G, with node 0 as V_{in} and node 6 as V_{out}.

Adleman's Approach of Molecular Computing:

The nondeterministic algorithm used by Adleman is shown below:

Step 1: Generate all possible random paths in G.

Step 2: Select those paths that start with V_{in} and ends with V_{out}.

Step 3: Extract itineraries that have correct length.

Step 4: Keep all paths that have all vertices at least once.

Step 5: If nonempty set is returned after **Step 4** then output is "Yes"; otherwise the output is "No".

Adleman performed massively parallel exhaustive random search strategy during his DNA Computing experiment. His entire wet lab simulation relied on some well know biotechnological operations. Though the instance of problems solved was very trivial but it revealed the computational power of DNA and provides a foundation for other models.

Adleman wisely used the parallelism property of DNA to implement Brute force approach i.e. all possible paths of the given graph were tried and invalid paths are filtered out. Hartmanis [35] showed that its impractical to solve a 200-noded HPP graph using Adlemans approach because it would require DNA of mass equal to 24 times the Earths size. Turing machine is a mathematical model which can compute any problem that can be represented in the form of an algorithm. It was proposed by Alen Turing in 1936. First DNA turing machine was proposed by Charles Bennett but it was based on imaginary restriction enzymes [36]. Later on in 1995 Paul Rothemund [37] proposed a practical implementable DNA Turing machine using the functionality of existing restriction enzymes. He encoded the transition table of a Turing machine in the form of DNA sequences and a series of ligation and restriction reactions were carried out on them to mimic the operations of Turing machine. The success of simulating Turing machine by DNA was significant as it puts DNA computing on equal footing with any other computational model. Reif et al. [38] established the turing-completeness of DNA Computation by adding extra splicing operation to the basic sets defined by Adleman.

Lipton [39,40] came up with an NP problem called *Satisfiability problem* (SAT). Adleman and Lipton demonstrated the probability of the use of DNA computing for solving NP problems like HPP and SAT problems. Their combined work is known

as **Adleman - Lipton Model**. A defined set of operation of the Adleman-Lipton model is stated below:

It is considered that there are 'n' numbers of test-tubes i.e. T_1, T_2,....., T_n consisting of DNA sequences or finite set of strings over alphabet $\sum$={ A, T, G, C }.

The basic steps of DNA-computing can be expressed as following bio-operations:

Extract: Provided a test tube T containing sequence S, this operation generates two tubes: one that had the required sequence represented as +(T_1, S) and other test tube consists of remaining sequences -(T_2, S).

Merge: Given two test tubes T_1 and T_2, the merge operation generates a new test tube T with the content of both without modifying.

Detect: Given a test tube T, this operation returns the logic value YES if there is at least a DNA molecule in it otherwise the logic value is read as NO.

Discard: Given a test tube T, this operation simply discards the content of the test tube.

Amplify: Given a test tube T; amplify (T, T_1, T_2) produces two new test tubes T_1 and T_2 that are identical copies of T.

Append: Given a test tube T and a sequence S; append (T, S) affixes S at the end of each sequence in T.

Amos, Gibbons and Hodgson Model:

Inspired by Adleman [1, 2] and Lipton [39, 40], Amos et al. [41] proposed a generalized parallel filtering model. Unlike Lipton's generalized model which only describes 3 SAT problem, Amos's model validated its generalizability by demonstrating several algorithms for NP complete problems like 3-vertex-colourability

problem, Hamiltonian path problem (HPP), the subgraph isomorphism problem, the maximum clique and maximum independent set problem. The operation set employed in the model are as follow:

remove(U, $\{S_i\}$): This operation was used to remove any string containing S_i as substring from a set U.

union($\{U_i\}$, U): This operation was used to create a set U obtained by union of sets U_i.

copy(U, $\{U_i\}$): This operation was employed to create several replicas, U_i, of set U.

select(U): This operation was used to check whether the set U is empty or not.

The complexity of each operation was assumed to be constant.

Table 2.1: Overall Complexity of some algorithms within Amos's model

Algorithm	**Complexity**
3-vertex-colourability problem	O(n) parallel-time where n = |V| where V is the set of vertices of graph.
Hamiltonian Path Problem (HPP)	Constant parallel-time given P_n where P_n is the set of all permutation of integers {1,2, ... ,n}.
The subgraph isomorphism problem	O($\|V_s\|$) parallel time where V_s is set of vertices in subgraph S
The maximum clique problem	O(|V|) parallel time where V is the set of vertices in the problem graph.
Maximum independent set problem	O(|V|) parallel time where V is the set of vertices in the problem graph.

Amos's model was different from other prior filtering model as it didn't use the separation step and the removed strands were discarded. This feature leads to

lack of cost effectiveness and non-reusability. In the following year a new approach called Sticker model was proposed with relative amount of reusability.

Surface Model:

Reactions on free floating DNA solution is very error prone as there is great chance of washing off the useful strands which eventually elevates the error rate. To avoid the loss of information and to ensure reusability, the DNA strands are immobilized on a surface like gold or glass. This ensures robustness during processes like purification and separation. The immobilized strand retains its normal property and undergo all the biochemical reactions like hybridization, denaturation etc. as before.

Associative Memory Model:

Eric Baum [42] build a large associative memory using a new constructive approach called Associative memory model. In this model the partial information of the data content can be used as an address of the data. To store a word, append the DNA subsequences corresponding to each of its components together to form a molecule. To retrieve a word given a cue, retrieve its associative molecule.

Sticker Model:

Sticker model ensures the reusability of DNA strands, hence got the attention of researchers. Unlike several prior models, it didn't require strand extension and also minimize dependence on enzymatic reaction. The central idea of the sticker model system is to store binary information in a more efficient way and allow having a random access memory. Information was encoded in the form of strings of DNA which in turn obtained by substrings of defined length. The encoded strand was allowed to undergo hybridization reaction with the complement of certain subsequence (sequence representing bit i) of the DNA strand in order to set bit i. On the other hand simple denaturation was done to reset the bit. Though the model is

theoretically promising but the practical implementation involves implementation difficulties.

Splicing Model:

The central operation used in splicing model is the action of restriction enzymes and exchange of specific DNA sequences between molecules. When restriction enzymes cuts at a recognition site (small sequence 4 to 6 nucleotides long), the DNA separates into two pieces. The sticky end produced during the cutting can be engineered to concatenate with another sticky end of some other DNA. This process is well known as crossover in genetic algorithm. The process can be briefed with an example.

Let S and T be two strands of DNA such that S = {a b} and T = {x y} where a, b, x, y are subsections of DNA. On action of splicing operation the two DNA strands are cut at restriction site and the prefix of S is concatenated with the suffix of T and similarly the suffix of S is concatenated with the prefix of T. The new DNA strand obtained is named as S' and T' such that S' = {a y} and T' = {x b}.

Frank Guarnieri and Carter Bancroft Model:

Frank Guarnieri and Carter Bancroft [43, 44] introduced a DNA-based addition algorithm. They encoded representation of all two digit integer in the form of a DNA sequence which can be added. Adding such a pair involves four steps, in which the appropriate complementary sequences link up and strands are successively extended to make new, longer strands, finally yielding the correct output. Input DNA sequences serve as successive templates for constructing an extended result strand. The novelty of their approach is the introduction of a place holder for the carry position while performing addition. A limitation of their approach is that the output strand of one operation cannot serve as the input strand for another round of addition.

One of the most important problems with these biological operations is the high error rate. A large variety of suggestions were proposed to overcome errors in DNA computing operations.

In the following years, several attempts have been made to solve numerous renown problems using DNA Computing approach: Boolean Satisfiability Problem (SAT) [45–47], Vertex Coloring Problem [48–50], Maximum Clique Problem [51], Vertex Cover Problem [52–54], Bin packing problem [55], set partitioning problem [56], time table problems [57–59] etc. Though initially DNA-computing approach employed to solve problems from the field of NP complete but it does not remain confined to that and spreads its application to diverse problems. DNA Boolean circuit simulation stole the limelight as it has the potential to be the building block of biological computer and also have several biomedical applications.

2.2 DNA Boolean Logic and Circuits

2.2.1 Classification of Boolean Circuits

Boolean circuits are an important Turing-equivalent model of parallel computing. Computational units which are responsible for carrying out information processing are called logic gates connected to each other by a network of inputs and outputs. The logic gates processes output on the basis of present inputs available at that moment and do not have any memory about the past. The size and the depth of the circuit are the standards to measure the complexity of any circuit. Boolean circuit can be categorized into three types depending on the inputs and outputs associated [60, 61]:

- *Unbounded fan-in Boolean circuit:* No limitation to the numbers of inputs to

both AND-OR gates.

- *Semi-unbounded fan-in Boolean circuit:* AND gate is limited to two inputs and there is unlimited number of inputs for OR gate.
- *Bounded fan-in Boolean circuit:* Both AND and OR gate have two inputs.

Several models have been proposed in the field of DNA computing to simulate either individual logic gates or Boolean circuit. Boolean circuit is a network of logic gates, output of a gate in previous level is fetched as input to the gate in the next level and so on until the gate at highest level is reached. Though several DNA computing models for independent logic gate is proposed but only a few models included the measure for entire circuit simulation. In the following section some of the models for logic gate and Boolean circuit simulation are discussed.

2.2.2 Ogihara and Ray Boolean Circuit Model

Bounded fan-in Boolean circuit could be visualized as a directed acyclic graph. The size of the circuit is measured in terms of total number of gates in the circuit and the depth is measured by the longest path from the leaf node to the root. The input nodes or leaf nodes have in-degree 0 and the intermediate nodes and output node have maximum in-degree two. Only the root node has out-degree 0. The leaf nodes are associated with input variables x_i and can be represented by set X=(x_1,x_2,.,x_n). Each gate node, g_i is associated with some Boolean function $f_i \in \Omega.\Omega$ is referred as circuit basis (illustrated in **Figure 2.2**).

Ogihara and Ray [4, 5] followed a constructive approach to emulate the first ever bounded fan-in Boolean circuit. By following constructive approach instead of filtering, Ogihara managed to overcome the constraint faced by Adleman in bigger problem size (200-node HPP graph using Adlemans approach would require

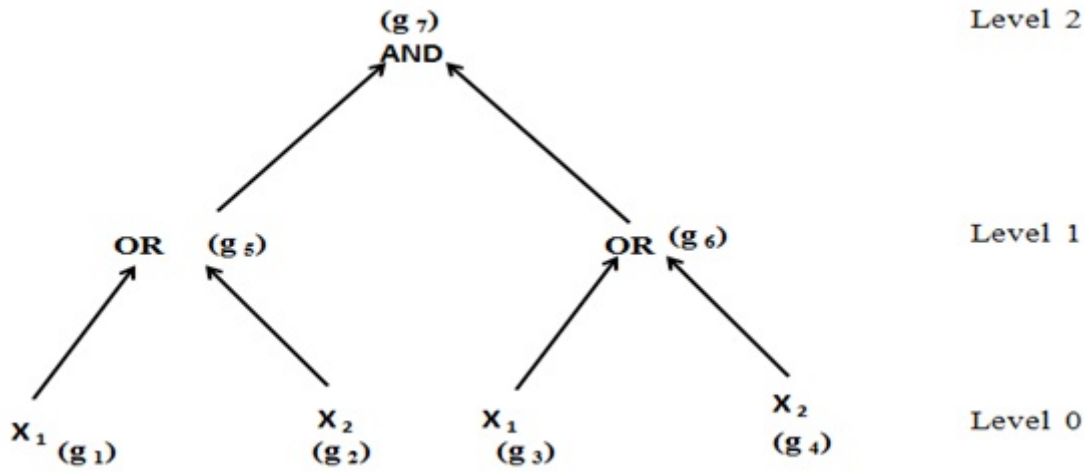

Figure 2.2: A bounded fan-in AND-OR Boolean circuit of size 7 and depth 2.

DNA of mass equal to 24 times the Earths size). **Figure 2.3** represents OR and AND gate simulation of the Ogihara's model.

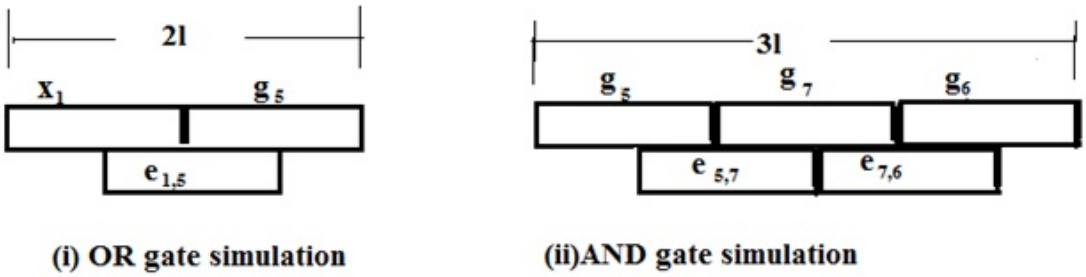

Figure 2.3: OR and AND gate simulation of the Ogihara and Ray experiment.

The complexity of Ogihara's model is proportional to its depth and maximum fan-out of the circuit. Amos and Dunne [3] proposed a NAND gate model with same complexity as Ogihara and Ray's model but with easy implementation. The work of Ogihara and Ray is significant as it is the first simulation model of Boolean circuit at molecular level and also it verifies Turing- completeness of DNA computers however the model was not successfully implemented. In the following years, some other properties of DNA such as finite splicing technique [6] and restriction enzyme [7] was proposed to simulate NAND gates but they suffer from the problem of

non-reusability as the restriction enzymes digested the gate strands after every operation. Also the model lack reliability as enzymatic reaction requires specific temperature, salt condition and co-factors. Wenbin Liu et al. [8, 9] demonstrated computation model based on chemically induced hairpin formation in the presence of Naphthyridine Dimer [62]. In **Section 2.2.3** the model proposed by Wenbin and his group is discussed.

2.2.3 DNA - NAND gate Model Based on Induced Hairpin Formation

Wenbin liu et al. [9] proposed a reusable NAND gate. The gate strand consisted of subsections which corresponded to two input variables i.e. x_i (i = 1, 2) each of length ℓ and two self-complimentary sequences represented by s and $\bar{s}$. The gate strands could be represented as 3'-s-x_2-x_1-$\bar{s}$-5'and immobilized to a surface at 5'end. The s and $\bar{s}$ sequences were of length 'm' with G.G mismatch. The computational mechanism of the model relied on the induced hairpin formation property due to self - hybridization of G.G mismatched nucleotides of s and $\bar{s}$ sequences in presence of naphthyridine dimer (ND). The sequences x_1 and x_2 formed the loop and s and $\bar{s}$ constitute the stem part of the hairpin.

As claimed by W. Liu et al. the theoretical model proposed by them had time complexity proportional to the depth of the circuit. However the model was not generalised and it demonstrated the simulation of only two gates ie. NAND and XOR.

Ahrabian et al. [61,63] published an algorithm to simulate DNA fan - in AND - OR Boolean circuit with time complexity $O(1)$. But he did not included any feature to evaluate circuit consisted of gates other than AND - OR. Mahnaz Kadkhoda [64, 65] proposed a NAND gate model with reduced number of passes in each

level. Due to biocompatibility, DNA gate finds new and demanding application in biomedical. Ehud Shapiro and Binyamin Gil [66] constructed a logic gate with the capability to automatically diagnose and release an anti-cancer drug within a cell on detection of certain carcinogenic behavior at cellular level. Frezza et al. [67] had demonstrated 2-input AND, OR, AND - NOT logic gates and claimed that the model could be scaled - up to form multi - level circuits, but the model suffered from the limitation in the size of circuit that could be simulated. In **Section 2.2.4** Zoraida et al. [10] model is critically discussed.

2.2.4 DNA Generalized Model to Evaluate Boolean Circuit

Zoraida et al. [10] proposed a novel algorithm to simulate any DNA Boolean operator. The model was capable to overcome some of the drawbacks of prior models, such as universality in representation of 0 and 1, applicability to simulate any kind of Boolean function, generalizibility, parallelism and reusability. The model employed only hybridization operation at each stage and apart from simulation of basic logic gates, it further emulated binary adder, subtractor, four bit carry ripple adder etc.

The final output of the algorithm is stored as elements of array consisting of combination of 0, 1 and *. The gate strand was then synthesized by replacing the values of 0s, 1s and blocker sequence "*" with the preassigned complementary strand. **Table 2.2** represents some of the designed gate strands.

Table 2.2: Output of above proposed algorithm

Gate	Gate Strand(3'-5')
OR	3'-$\bar{1}$-$\bar{1}$-$\bar{0}$-$\bar{1}$-5'
AND	3'-$\bar{1}$-$\bar{1}$-5'
NAND	3'-$\bar{0}$-$\bar{0}$-$\bar{1}$-$\bar{0}$-5'
NOR	3'-$\bar{0}$-$\bar{0}$-5'
EX-OR	3'-$\bar{0}$-$\bar{1}$-$\bar{0}$-5'
EX-NOR	3'-$\bar{0}$-$\bar{0}$-*-$\bar{1}$-$\bar{0}$-5'
NOT	3'-$\bar{0}$-5'
BUFFER	3'-$\bar{1}$-5'

The digital equivalent inputs to the model were fetched in the form of DNA molecular beacon with loop portion consisting of sequence I_1 and I_2. During execution, the input MBs were exposed to the surface and allowed to undergo hybridization. If they anneal successfully than the fluorescence occurred (output was read as '1') otherwise as '0'.

The working principal of the proposed model is demonstrated in **Figre 2.4**.

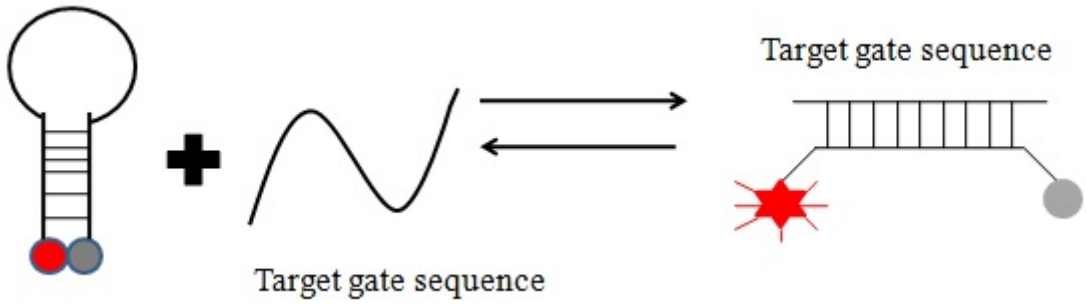

Figure 2.4: Emission of fluorescence signal on hybridization of MB and target gate sequence.

Though Zoraida and her group demonstrated the theoretical simulation, only a few gates and combinational circuits such as half-adder, full-adder, four-bit carry ripple adder, half subtractor, full subtractor and four-bit borrow ripple subtractor but they claimed that their algorithm is applicable to simulate any kind of logic gate. Despite of inclusion of several features the implementation cost of the model was high as four sets of non reusable MBs were used as inputs. Also the maximum number of input variables allowed was limited by the length of the loop of MB.

Park at al. [68] proposed AND, OR and YES gates using DNA replication as tool of computation and also claimed that their model could be used as sensors for detection of some metal ions. Goel and Morteza [69] introduced a time-responsive generalized model to implement any Boolean operation. A thermodynamically and kinetically reversible DNA Boolean circuit model was reported by Genot et al. [70]. They claimed that the proposed model had processing power to continuously re-compute their outputs depending on the changes in the inputs. DNA strand displacement property was used by Wei Li et al. [71] to implement three-input switchable gate at molecular level. They experimentally showed that a multifunctional circuit could be obtained by integrating two three-input majority gates in serial fashion. Most recently the property of DNA strand displacement reaction was employed by Wei Li et al. [72] to reproduce XOR and AND gates at molecular level and were integrated to realize half adder logic circuit.

2.2.5 Chemically Induced Logic Gate

The efficiency of most of logic gate simulation model is limited by the involvement of too many erroneous and time consuming biochemical processes. As the gap between the chemical science and molecular technology is bridging, the DNA based simulation and processing of logic gates is adopted by researchers in the field of chemistry as a fascinating area of research. Recently several advances have been

reported to develop simple DNA logic gates using several features like luminescence, colorimetric, electrochemical or electrochemiluminescence signals responding to biomolecular reactions for sensory and computing applications [73]. Led by the work of Stojanovic and other contemporary researchers several properties and structural specifications of DNA such as G-quadruplexes, i-motif, DNAzymes and aptamers are incorporated into the design of basic Boolean gates [74–77]. The advantage of such chemically induced gates lies in its fast response time, controllability and reusability feature. Recently secondary structure of DNA such as G-quadruplex and i-motif structure finds its way into several gate simulation models due to their unique properties such as highly specific binding properties, polymorphic versatility, and self-assembly. There are several models reported where i-motif structure co-existed with G-quadruplex [78, 79], but such models witness the setback in its complex maintenance and high cost. In few models G-quadruplex structure is solely used whereas in some other models only i-motif structure is used in construction of gate designing [80–83]. Yunhua Shi et al. [22] demonstrated DNA logic gates using only i-motif structural induction in response to the presence of H^+, Ag^+, and I^- as inputs. This simple technology has potential to simulate the functionality of OR and INHIBIT gate. Henry Albert Day et al. [84] published the switch ability of pH induced i-motif structure to hairpin and vice-versa. During the experiment they initially induced i-motif structure in human telomeric sequence at pH 5.5 which was again altered to hairpin structure by adding Cu^{2+} at room temperature without changing the pH. Further they continue their experiment where they successfully reverse the hairpin structure to i-motif structure by adding EDTA (Ethylenediaminetetraacetic acid).

Recently, it was found that i-motif can also form at neutral pH with Ag^+ ion owing to the strong interaction of Ag^+ with cytosine to form a C-Ag^+-C complex. Inspired by the response of i-motif DNA in presence of inputs H^+, Ag^+ Yunhua Shi and his group had developed a simple platform technology to emulate several logic

gates such as OR and INHIBIT. They designed a fluorescence probe named DMSB (2,2'-diethyl-9-methylselenacarbocyanine bromide) to detect the i-motif formation which further used as output of logic gates. The i-motif forming sequence plays a very important role in the entire experiment hence the complementary sequence of aptamer AS1411 (cAS1411) was chosen because its motif structure caused more obvious variance of the probes fluorescence than other oligonucleotides. The gate simulation of OR and INHIBIT gate is shown below in **Figure 2.5**. In 2017 a modular circuit design strategy was proposed by Chatterjee et al. [85] by adding several advanced features to DNA computing field. They demonstrated a faster and reusable DNA circuit whose internal kinetics was independent of concentration that was obtained by localized intermolecular reactions. The logic gate and signal transmission lines were created by spatially arranging reactive DNA hairpins on a DNA origami. All the components in the solution simultaneously perform same task in parallel fashion and could not individually perform different computation on its own.

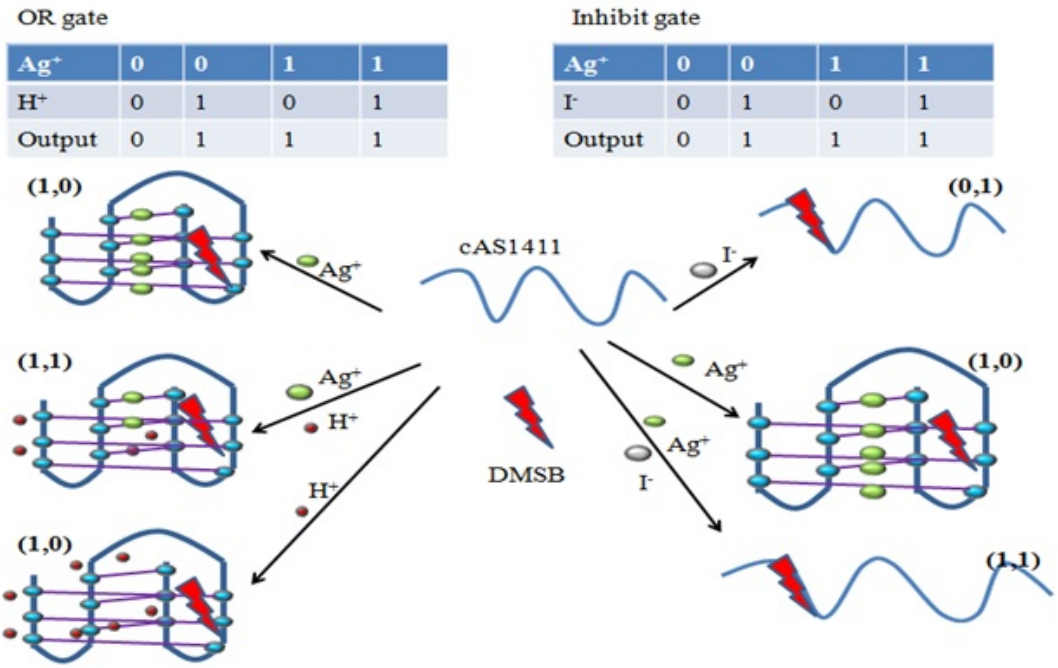

Figure 2.5: Demonstrates the structural switching of cAS1411 induced by Ag^+, H^+, and I^- and the fluorescence variance of DMSB.

2.2.6 Readout Mechanism

Reading the output is a crucial stage of any gate design model. Since the entire operation is carried out at molecular level hence efficient mechanism has to be integrated to transduce the biological signal to appropriate user understandable signal. During selecting the proper readout mechanism certain factors needed to be taken care, such as; accuracy, cost effectiveness, fast response time. Selection of proper output mechanism greatly affects the overall efficiency of the model. Each of the readout technique has certain advantages as well as disadvantages. In the following table **Table 2.3** the overview of some honorable models in the field of DNA computing and logic simulation over the years along with the readout mechanisms are shown.

Table 2.3: Overview on DNA based models in the last two decades.

Year	**Type of DNA Computing Model**	**Complexity**	**Reference**	**Readout Mechanism**
1994	First DNA Synthetic Computer (Hamiltonian path problem)	-	L. Adleman [1]	Graduated PCR
1995	DNA Solution of Hard Computational Problems(SAT Problem)	-	Lipton [39]	The non empty last test tube after operation of all clauses was read as satisfiable otherwise non satisfiable.
1996	Implementation of first DNA Turing machines	-	P.W.K. Rothmund [37]	Transition table is encoded in DNA olionucleotide sequences and state-symbol transition operation was performed by the function of restriction and ligation enzymes.

Continued on next page

Table 2.3 – continued from previous page

Year	Type of DNA Computing Model	Complexity	Reference	Readout Mechanism
1997	First DNA Boolean circuit simulation(AND-OR circuit simulation)	Runtime is proportional to logarithm of maximum fan out of the circuit	Ogihara and Ray [4]	**For AND gate:** the presence of strand of 3ℓ length is considered as output 1. **For OR gate:** the presence of strands of length 2ℓ evaluates the output as 1. **Output:** visualized by autoradiography on an X - ray film.
1997	M. Amos and P. Dunne(Boolean circuit simulation)	Run time proportional to depth of the circuit.	Amos et al. [3]	The output of gate t_j is read as 1 or 0 on the basis of presence or absence of string of length $u_j + 2\ell$ in the gel. Output: visualized by running in gel.

Continued on next page

Table 2.3 – continued from previous page

Year	Type of DNA Computing Model	Complexity	Reference	Readout Mechanism
2004	First parallel model for circuit simulation(AND-OR circuit simulation)	O(1)	Ahrabian et al. [61]	The presence of any single stranded sequences that starts with δ_S and ends with δ_T in the final test tube is evaluated as 1. **Output:** amplification by PCR and extraction sequence starts with δ_S and ends with δ_T
2004	A new DNA computing model for the NAND gate based on induced hairpin formation	Time complexity proportional to the level of the circuit	Wenbin Liu et al. [9]	Hairpin structure in presence of ND is evaluated as 1 other wise as 0. **Output:** strand configuration was detected by surface Plasmon resonance (SPR) imaging.

Continued on next page

Table 2.3 – continued from previous page

Year	Type of DNA Computing Model	Complexity	Reference	Readout Mechanism
2009	First generalized model (simulation of any logic function)	Proportional to size of circuit	Zoraida et al. [10]	Success of MB hybridization with target subsequence of gate strand is read as 1 otherwise as 0. **Output:** Flouriscence emission of MB.

Continued on next page

Table 2.3 – continued from previous page

Year	Type of DNA Computing Model	Complexity	Reference	Readout Mechanism
2010	Metal ion triggered DNA logic gate.	-	Park et al. [68]	**AND gate:** In presence of both metal ions i.e. Hg^{2+} and Ag^{+} non natural base pairing is facilitated between T-T and C-C mismatches(T-Hg^2+-T and C-Ag^{+}-C) which leads to successful execution of exponential PCR which reads as 1. **OR gate:** In presence of metal ion Hg^{2+} , T-Hg^{2+}-T complex is formed which trigger the polymerase activity to successful execution of PCR (read as 1). **Output:** fluorescence emission after 30^{th} PCR cycle.

Continued on next page

Table 2.3 – continued from previous page

Year	Type of DNA Computing Model	Complexity	Reference	Readout Mechanism
2011	Reversible logic circuit	-	Genot et al. [70]	A robust, reversible, input dependent AND gate based circuit. **Output:** Fluorescence signal of designed secondary structure is read as 1 otherwise as 0.
2014	DNA logic gates utilizing a H^+/Ag^+ induced i-motif structure	-	Yunhua Shi et al. [22]	**OR gate:** Induction of i-motif in presence of either Ag+ or H+ ions. **Inhibit gate:** Induction of i-motif only in presence of both Ag^+ or I^- ions. **Output:** Fluorescence probe named DMSB is used to detect the i-motif.

Continued on next page

Table 2.3 – continued from previous page

Year	Type of DNA Computing Model	Complexity	Reference	Readout Mechanism
2016	DNA based arithmetic function: half adder	-	Wei Li et al. [72]	Initially XOR and AND gate were designed and used as building block for half adder realization based on strand displacement property of DNA. **Output:** Due to opening of hairpin the fluorescence intensity increase which is read as 1 otherwise as 0.

Continued on next page

Table 2.3 – continued from previous page

Year	Type of DNA Computing Model	Complexity	Reference	Readout Mechanism
2017	Implementation of cascade logic gate	-	Jinting Gao et al. [86]	Cascade logic gate is successfully implemented such as AND-OR-INH (INHIBIT), AND-INH and OR-INH. **Output:** Hemin integrated colorimetric system (TMB/H2O2/ Hemin) was used as logic platform. Change in color of the solution to yellow in presence of either two or three of the G-DNA, K^+ and Cu^{2+} was used as operation mechanism. Similarly the strong inhibitory property of antioxidant Tertiary Butyl Hydroquinone (TBHQ) was also used in gate operation.