IJCTA, 9(21), 2016, pp. 281-287
© International Science Press

DNA Computing Model for Realization of Boolean Circuit

Kuntala Boruah* Jiten Ch.Dutta**

***Abstract* :** DNA computing is an emerged paradigm bridging the gap between computer science and molecular biology. In this paper an attempt has been made to simulate Boolean circuit using computational property of Deoxyribonucleic Acid (DNA). The proposed model uses standard biochemical reactions such as sequence-specific annealing, ligation, separation by affinity purification as computational tool. This theoretical model contributesin enhancing the flexibility of computable Boolean circuit by allowing different kind of logic gates in each level. A uniform standard is proposed while designing the gate strands. Specially designed fluorescence labeled probe i.e.Molecular Beacons (MB) is used as input to the Boolean circuitand accordingly entire circuit operation is carried out at molecular level. Special care has been taken to exploit the parallelism property of DNA. The simulation of all the gates in any single level is executed in one test tube therefore the time complexity is proportional to the depth of the circuit.

***Keywords* :** Boolean circuit, DNA,Biochemical reactions,Molecular Beacons, Annealing, Ligation, Affinity purification, Logic Gate.

1. INTRODUCTION

In 1994 Adleman[1] published a paper proposing an algorithm to solve a seven vertex Hamiltonian path problem using DNA computing approach. His experiment introduced a new paradigm of immense possibility. Since then several researchers are publishing their works by applying DNA Computing approaches in solving wide range of problems.Ogihara and Ray[2] proposed an algorithm to simulate AND -OR base Boolean Circuit with run time proportional to size of circuit. Amos el al.[3] proposed an improved NAND based circuit with run time proportional to depth of circuit.But both of the models use error prone technique such as PCR.Since then several AND-OR based Boolean circuit has been proposed. Erik Winfree and Katrin [4] exploit finite splicing system to simulated Boolean circuit. Mulawka et al. [5] published his work in 1999 to simulate DNA-NAND gate using restriction enzyme as key operator but the model lakes reusability. A fan-in Boolean circuit was developed by Ahrabian and Nawzari [6]. Contemporary to those, Liu et al. [7] published a theoretic NAND Gate simulation model using the induced hairpin formation of DNA in presence of G-G mismatch. MahnazKadkhoda [8] proposed a NAND gate model as an improvement to the existing model by reducing number of passes in each level. Ehud Shapiro and Binyamin Gil [9] constructed a DNA logic gate which is capable of automatically diagnose and release an anti-cancer drug within a cell.Recently Christy M.Gearheart et al. [10] and Zoraida et al. [11] proposed models to simulate logic gate.

The advantage of this proposed model is in its ability to solve Boolean circuit with different types of gates simultaneously in single level.Most of the earlier models suffered from the constraints that each level of the circuit requires the gates to be of same type but the author of this model claims to overcome thislimitation by allowing any kinds of gate in a single level. Also the model increases parallelism and decreases human intervention to a great extant. Since level wise simulation is done, the number of test tubes required is proportional to the depth of the Boolean circuit.

* Department of Electronics and Communication Engg, TezpurUniversity, Assam, India Kuntala17@gmail.com

** Department of Electronics and Communication Engg, TezpurUniversity, Assam, India

Organization of rest of the paper is as follows: Section II describes the DNA computing principles. Section III gives a brief introduction of Boolean circuits. Section IV presents the complete simulation process for computing the given Boolean Circuit. The paper concludes with Section V.

2. DNA COMPUTING PRINCIPLES

DNA (Deoxyribonucleic acid) has predictable powerful computation property which make it as favorite choice as a computation tool. Several computational models are proposed to harness the power of this genetic biomolecules. Strand of DNA is a polymer of four kinds of nucleotides: Guanine (G), Thymine (T),Adenine (A) and Cytosine(C). A, T and G, C are Watson-Crick complementary pairs and this complementarily is predictable as well as robust to a great extant. A single stranded DNA in presence of its complementary strand establishes hydrogen bonds between their nucleotides and obtains its stable double helix structure. Inany digital system information is encoded in the form of finite string of binary digit over alphabet {0, 1}.Similarly, DNA strand also can be stated as a finite string of alphabet over four nucleotides $\sum$ ={T, A, C, G}.

Biochemical tools used in the proposed model : In this model the author uses few well organized wet lab processes to induce biochemical changes to the nucleic acid chain:

Synthesis : Designing and restructuring of oligonucleotide sequences is called sequence synthesis. In this model several unique DNA sequences are needed to be designed to encode each of the variable notations and for tagging the gate numbers(shown in Table3).

Hybridization :DNA has its inherent tendency to form double stranded helical structure obtained by establishing hydrogen bond between the complementary nucleotides. During the hybridization process Watson-Crick base pairing is always followed.

Denaturation : On rising the temperature the free energy of strands of DNA rises as the hydrogen bonds between the two strands is broken.This process of splitting the double stranded DNA into single stranded is reverse of hybridization and is called as denaturation process.

Ligation : Ligation is process in which phosphodiester bond is established between consecutive nucleotides in presence of enzyme called ligase. In this process two single strands of DNA are tailored into one single strand of DNA.

Affinity Purification : Affinity purification is a process to extract and separate a particular DNA strand of interest. In this method the complement of the tag sequence is used to fish out the target DNA strand.

3. BOOLEAN CIRCUIT

Boolean circuits can be stated as generalization of Boolean formulas. A Boolean circuit consists of network of logical gatesresponsible for executing the Boolean functions.It is categorized as follows: (*i*) Unbounded fan-in (*ii*) semi-unbounded fan-in (*iii*) bounded fan-in. In Graph theory point of view, it can be stated as acyclic directed graph, G (V, E).Input nodes have in-degree 0, intermediate gate nodes have maximum in-degree 2 and the last level *i.e*. output gatehas out-degree 0. Inputsarerepresented with a Boolean Variable X and Y and the gate nodes g_i are associated with Boolean function $f_i \in \Omega$, where Ω is the circuit basis. Complexity of any Boolean circuit can be defined in terms of its size or depth. A simple Boolean circuit is shown below:

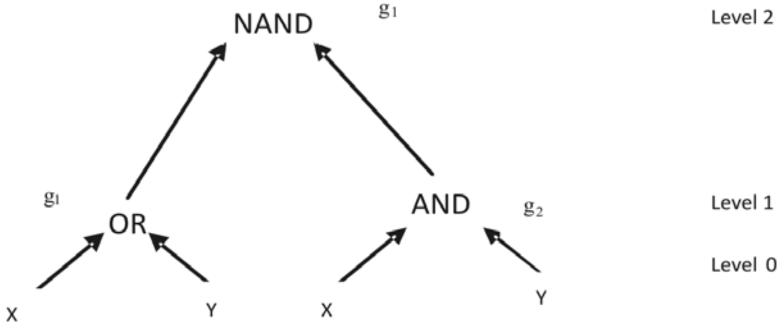


Fig. 1. AND-OR-NAND Boolean Circuit of Depth 2.

The inputs, gate operator and outputs are encoded in the form of specially designed DNA strands and executed in the form of molecular operation. All the desired biochemical operations are carried out in test tubes.

4. COMPUTATIONAL MODEL FORSIMULATION OF BOOLEAN CIRCUIT

A Boolean circuit consists of k level (0,1….., $k-1$) where level 0 is the input level and $k-1$ level is the output level. Operation in each of the levels is executed in the form of biochemical reaction in separate test tubes *i.e.* in a k leveled Boolean circuitthere are k test tubes involved. All the intermediate operations are performed at the level 1 to level k-2 and test tube of level k gives the output.

To simulate any Boolean circuit,the first step is to encode each individual logic gate in the form of DNA strands. While designing these strands a uniform technique is to be followed. Each gate strand is designed on the basis of its corresponding truth table.In the procedure the rowsof the truth table arescannedone by one and depending on the output value, gate strand is encoded. Table 1 shows a general form of a truth table with X and Y as input variables and Z_i as output.The Step 1, Step 2 and Step 3 are gate encoding technique and Step 4 and Step 5 are Boolean circuit execution step. All the stepsinvolved in simulating Boolean circuitcan be summarized as follows:

Table 1. Truth table with two inputs

X	Y	Z_i
0	0	Z_1
0	1	Z_2
1	0	Z_3
1	1	Z_4

Step 1: In this step variable notations are inserted into an array named “gate_design”.Once the array is obtained the required DNA strand is obtained by replacing the elements with the pre-assigned strands of DNA. Insertion into arraygate_design [ ] can be summarized as follows.

Scan the rows of truth table sequentially until $i <= 2^n$. ‘i’ is used for counting the rows to be scanned in the truth table *i.e.* $i = 1, 2……2^n$ if n is number of variable.

- If Z_i evaluates to 1 in the truth table, then check the values associated with the variable X and Y. If X is 1 then enter X^1 as an element into the array gate_design[index++] otherwise enter X^0, similarly Y^1 or Y^0 is entered as the next element depending on the values of the variable in the row.
- If Z_i evaluates to 0, then complements of variable notations inserted into the array. *i.e.* if X is 1 then enter $\overline{X}^1$ as element in the array gate_design[]otherwise enter $\overline{X}^0$, similarly $\overline{Y}^1$ or $\overline{Y}^0$ is entered as next element. After every insertion the index of the array is incremented *i.e.* index = index + 1.

  “gate_design” is the array where the variable notations are stored according to the output value of the corresponding truth table.Insertion into array is continued until end of the table is reached. Table 2 shows the gate_design[] of few gates.

Table 2. gate_design[] for few logic gates.

Gate	*Gate_Design[ ]*
OR	$\{ \overline{X}^0\ \overline{Y}^0, X^0, Y^1, X^1, Y^0, X^1, Y^1 \}$
AND	$\{ \overline{X}^0, \overline{Y}^0, \overline{X}^0, \overline{Y}^1, \overline{X}^1, \overline{Y}^0, X^1, Y^1 \}$
NOT	$\{ X^0, \overline{Y}^1 \}$

Gate	*Gate_Design[ ]*
NAND	$\{X^0,Y^0, X^0,Y^1,X^1,Y^0, \overline{X}^{1}, \overline{Y}^{1}\}$
XOR	$\{ \overline{X}^{0}\ \overline{Y}^{0}, X^0,Y^1, X^1, Y^0, \overline{X}^{1}, \overline{Y}^{1}\}$
XNOR	$\{X^0,Y^0, \overline{X}^{0}\ \overline{Y}^{1}, \overline{X}^{1}, \overline{Y}^{0}, X^1, Y^1\}$
NOR	$\{ X^0,Y^0, \overline{X}^{0}, \overline{Y}^{1}, \overline{X}^{1}, \overline{Y}^{0}, \overline{X}^{1}, \overline{Y}^{1}\}$

Step 2 : Designing ofgate strand

Since a Boolean circuit has different gates in each level and all the gate strands of a particular level are poured into the same test tube, therefore, to distinguish one gate from another a unique tag strand is appended to the gate_design[] array. Tag sequence is actually a DNA sequence especially reserve to identify gate numbers.In this paper a Boolean circuit with maximum two gates in a level is simulated, so only two tag strand is sufficient.In level 1 tag sequence for first gate is represented by $\text{tag}_1$and second gate is represented by tag_2.Similarly, in level 2 there is only one gate therefore $\text{tag}_1$is sufficient. Special care must be taken to retain the uniqueness of tag sequence and to avoid any possiblerepetition of the tag sequence in any of the pre-assigned sequences of variable notations. By appending $\text{tag}_1$and $\text{tag}_2$to the end of the “gate_design” sequence the final gate strand is obtained.

For Boolean circuit represented in Figure 1 the final gate strandsare shown below

Level 1

Gate 1: g_1(OR) **Gate 2:** g_2(AND)

$3'$—Tag_1— $\overline{X}^{0}, \overline{Y}^{0}, X^0,Y^1, X^1, Y^0, X^1, Y^1$—$5'$

$3'$—Tag_2— $\overline{X}^{0}, \overline{Y}^{0}, \overline{X}^{0}, \overline{Y}^{1}, \overline{X}^{1}, \overline{Y}^{0}, X^1, Y^1$—$5'$

Test tube 1 *i.e*. T1 contains strands of g_1 and g_2.The DNA gate sequence is represented as:

Similarly in Level 2 one NAND gate is present (Shown in Figure 1) and accordingly the gate strand has to be encoded. gate_design[] = $\{X^0,Y^0, X^0, Y^1, X^1,Y^0, \overline{X}^{1}, \overline{Y}^{1}\}$

Level 2

Gate 3 : *g*1 (NAND) : $3'$ —Tag_1—$X^0, Y^0, X^0, Y^1, X^1, Y^0, \overline{X}^{1}, \overline{Y}^{1}$—$5'$

Test Tube, T2 contains g_1.

Step 3: Final DNA gate strand.

Once the gate strands are designed the last step in strand design is to replace the elements in the encoded array by the corresponding pre-assigned unique DNA sequence as shown in Table 3. The strand obtains at the end is the desired DNA gate stand with orientation $3' \rightarrow 5'$(shown in Table 3). In Table 3each of the variable notation X^1, X^0, Y^1, Y^0 has been allotted a unique singlestranded $3' \rightarrow 5'$ oligonucleotide sequence and its complementary variables $\overline{X}^{1}$, $\overline{X}^{0}$, $\overline{Y}^{1}$ and $\overline{Y}^{0}$ are assigned with complements of assigned sequences with $5' \rightarrow 3'$ orientation.Figure 2 and Figure 3 shows the encoded Boolean circuit and DNA equivalent Boolean circuit.

Table 3. DNA Sequence assignmentto variables

Symbol	*$3' \rightarrow 5'$*	*Complements*	*Complementary $5' \rightarrow 3'$*
X^1	TCTGAC	$\overline{X}^{1}$	AGACTG
X^0	GCTATT	$\overline{X}^{0}$	CGATAA
Y^1	GTAACA	$\overline{Y}^{1}$	CATTGT
Y^0	GACTTA	$\overline{Y}^{0}$	CTGAAT

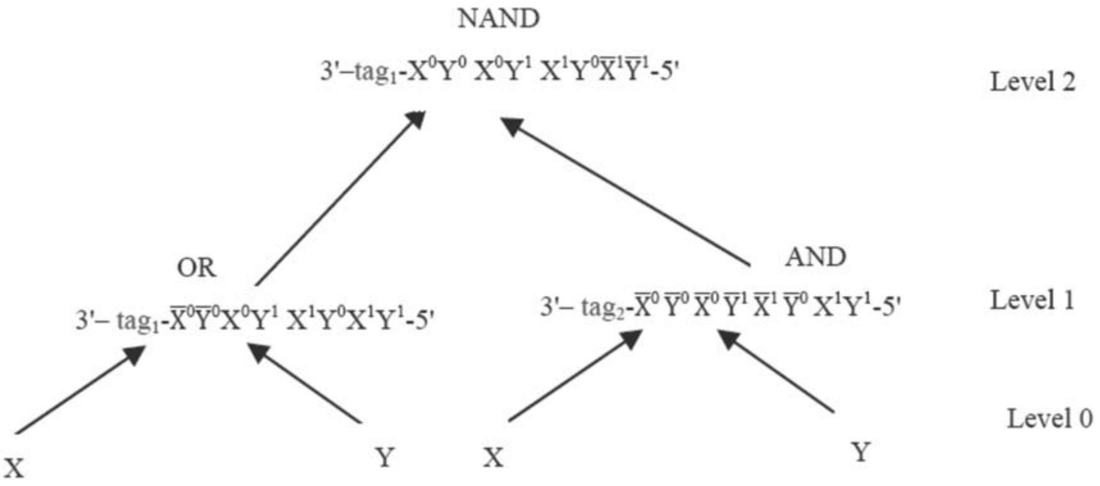

NAND
$3'\text{–}tag_1\text{-}X^0Y^0\ X^0Y^1\ X^1Y^0\overline{X}^1\overline{Y}^1\text{-}5'$
Level 2
OR
$3'\text{–}\ tag_1\text{-}\overline{X}^0\overline{Y}^0X^0Y^1\ X^1Y^0X^1Y^1\text{-}5'$
AND
$3'\text{–}\ tag_2\text{-}\overline{X}^0\,\overline{Y}^0\,\overline{X}^0\,\overline{Y}^1\,\overline{X}^1\,\overline{Y}^0\,X^1Y^1\text{-}5'$
Level 1
Level 0
X
Y
X
Y

Fig. 2. Encoded Boolean circuit.

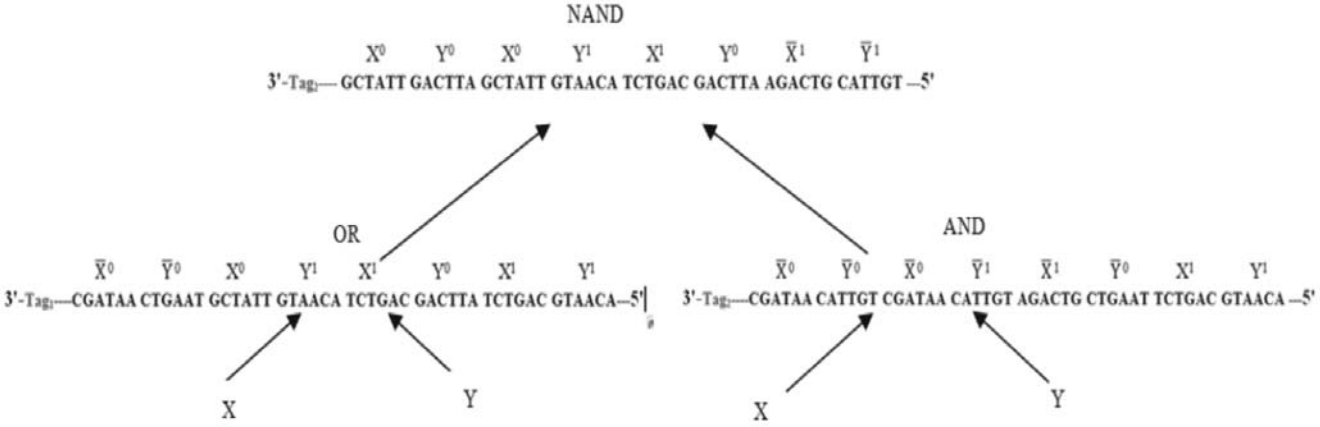

NAND
X^0 Y^0 X^0 Y^1 X^1 Y^0 $\overline{X}^1$ $\overline{Y}^1$
3'-Tag₂— GCTATT GACTTA GCTATT GTAACA TCTGAC GACTTA AGACTG CATTGT –5'
OR
$\overline{X}^0$ $\overline{Y}^0$ X^0 Y^1 X^1 Y^0 X^1 Y^1
3'-Tag₁—CGATAA CTGAAT GCTATT GTAACA TCTGAC GACTTA TCTGAC GTAACA–5'
AND
$\overline{X}^0$ $\overline{Y}^0$ $\overline{X}^0$ $\overline{Y}^1$ $\overline{X}^1$ $\overline{Y}^0$ X^1 Y^1
3'-Tag₂—CGATAA CATTGT CGATAA CATTGT AGACTG CTGAAT TCTGAC GTAACA –5'
X
Y
X
Y

Fig. 3. DNA equivalent Boolean circuit.

Step 4: Input Design

Input to the Boolean circuit has to be provided in the form of Molecular Beacons (MB). MB is actually used to sense the success of hybridization by attaching fluorophore and a quencher in its two ends of DNA hairpin. It is a single stranded DNA with complementary nucleotides in its 3' and 5' ends called as stem part; hence due toself-hybridization,they attains a structure as shown in Figure4. When the MB is in closed state it remains dark as the energy emitted by fluorophore is absorbed by quencher. But if MB finds its target sequence, the loop part hybridizes and result its loop to open up.During this fluorophore and a quencher moves away resulting transition from dark to bright. In this model inputs are encoded into the loop potion as complements of X and Y.

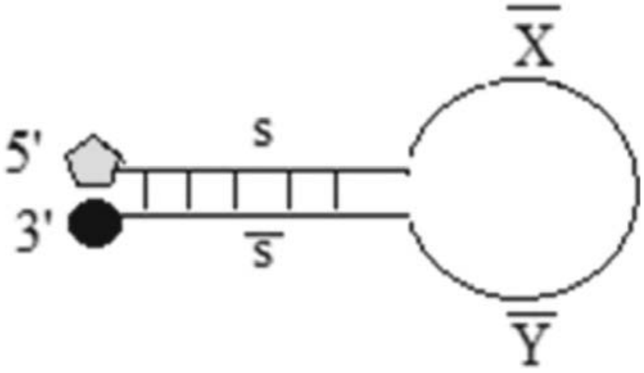


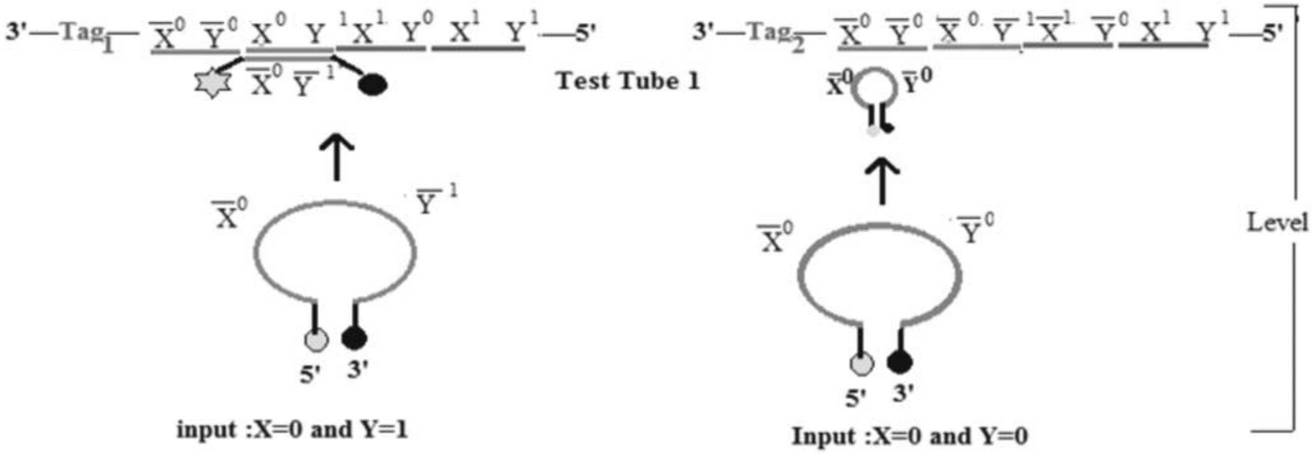
Fig. 4. Input strand

Table 4. DNA InputSequence

Inputs X,Y	*Derived Input*	*Input DNA Sequence*
0,0	$\overline{X}^0, \overline{Y}^0$	5' *s*–CGATAACTGAAT– $\bar{s}$ 3'
0,1	$\overline{X}^0, \overline{Y}^1$	5' *s*–CGATAACATTGT– $\bar{s}$ 3'
1,0	$\overline{X}^1, \overline{Y}^0$	5' *s*–AGACTGCATTGT– $\bar{s}$ 3'
1,1	$\overline{X}^1, \overline{Y}^1$	5' *s*–AGACTGCATTGT– $\bar{s}$ 3'

Step 5: Execution

Level 1 : Once the gate and the input strands are designed, the next phase is the execution phase.Test tube T1 consists of DNA gate strands corresponding to g1 and g2 of level 1.Input strands(MB) are poured into the test tube. In the presence of the tagret sequence, the input MB hybridizes with the gate sequence.Successful hybridization results in emitting fluorescence signal and read as 1.



3'—Tag1— $\overline{X}^0$ $\overline{Y}^0$ X^0 Y^1 X^1 Y^0 X^1 Y^1—5'
$\overline{X}^0$ $\overline{Y}^1$
Test Tube 1
3'—Tag2— $\overline{X}^0$ $\overline{Y}^0$ $\overline{X}^0$ $\overline{Y}^1$ $\overline{X}^1$ $\overline{Y}^0$ X^1 Y^1—5'
$\overline{X}^0$ $\overline{Y}^0$
$\overline{X}^0$
$\overline{Y}^1$
5' 3'
input :X=0 and Y=1
$\overline{X}^0$
$\overline{Y}^0$
5' 3'
Input :X=0 and Y=0
Level 1

Fig. 5. DNA reaction in Test Tube 1 for Level 1 with inputs (0,1) for OR gate and (0,0) for AND gate.

Since the test tube T1 contain both g_1 and g_2, affinity purification is carried out to separate the strands of both gates to check the output of each gate saperately. This is carried out by taking the complementsof 3' tag$_1$ and 3' tag$_2$ sequence to fish out the gate strands from the solution oftest tube.After separating different gates strads, flourescence emission is checked for each gate. If the fluorescenceemissionis detected than for that particular gate the output is read as one otherwise as zero.

For simplicity of explaination the Boolean circuit simulation process is shown for inputs (0,1) to OR gate and (0,0) to AND gate.Depending on output of g_1 and g_2 of Level 1 the input to next level is designed *i.e.* output of gates of previous level is encoded into the loop portion of input MB to next level.If output of g_1 is 1 then $\overline{X}^1$ sequence is considered in designing the input otherwise $\overline{X}^0$ is considered.Similarly depending on output of g_2, $\overline{Y}^1$ or $\overline{Y}^0$ is considered as next element of the input sequence.The output of the OR gate is read as 1(fluorescence emission: $\overline{X}^1$) and for AND gate the output is 0(no emission: $\overline{Y}^0$).The input to the next level is designed by ligating the outputs of the previous level

After designing the input to the next level,the input strand is poured to test tube 2 allowing them to hybridize.If fluorescence is reported in test tube T2 than the final output of the Boolean circuit is read as 1.In this example the MB hybrized with the sequence of NAND gate hence the final output is detected as 1. Figure 6 shows the complete simulation reactions.

5. CONCLUSION

In this paper a model is proposed to simulate DNA based Boolean circuit. The theoretical model involves standard biochemical reactions such as sequence-specific annealing, ligation, separation by affinity purification as a computational tool and avoids using error prone processes like PCR. The hybridization-induced fluorescent emission is observed while reading out the output. Unlike several existing methods, this model provides the flexibility of having different types of gates simultaneously in each level. Though this model seems to be theoretically sound, but we acknowledge that the physical realization in the laboratory involves practical difficulties. Future works must be carried out emphasizing the implementation aspects.

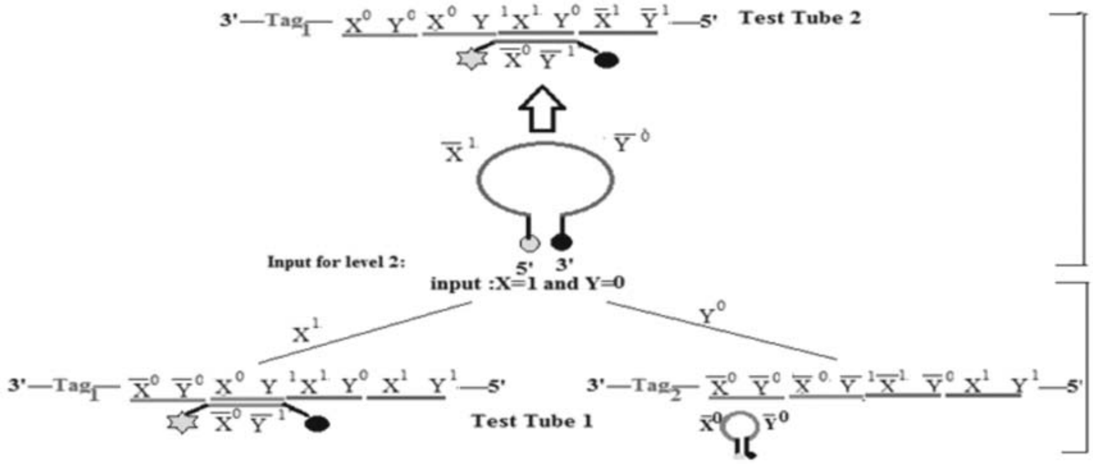

3'—Tag1— X^0 Y^0 X^0 Y^1 X^1 Y^0 $\overline{X}^1$ $\overline{Y}^1$—5'
Test Tube 2
$\overline{X}^0$ $\overline{Y}^1$
$\overline{X}^1$
$\overline{Y}^0$
Level 2
Input for level 2:
5' 3'
input :X=1 and Y=0
X^1
Y^0
Level 1
3'—Tag1— $\overline{X}^0$ $\overline{Y}^0$ X^0 Y^1 X^1 Y^0 X^1 Y^1—5'
$\overline{X}^0$ $\overline{Y}^1$
Test Tube 1
3'—Tag2— $\overline{X}^0$ $\overline{Y}^0$ $\overline{X}^0$ $\overline{Y}^1$ $\overline{X}^1$ $\overline{Y}^0$ X^1 Y^1—5'
$\overline{X}^0$ $\overline{Y}^0$

Fig. 6. DNA simulation of Boolean Circuit

6. ACKNOWLEDGEMENTS

The authors like to extend their gratitude to the Tezpur University for providing necessary facilities.

7. REFERENCES

1. L.Adleman, “Molecular computation of solutions to combinatorial problems” ,Science 266, pp.1021-1024,1994.
2. M.Ogihara,A.Ray,“DNA-based self-propagating algorithm for solving bounded-fan-in Boolean circuits” Genetic Programming , pp.725- 730, 1998.
3. M.Amos, P.E.Dunne, and A.Gibbons,“DNA simulation of Boolean circuits”, *in Proceedingsof 3rd Annual Genetic Programming Conference*, pp. 679-68, 1997.
4. K .Erik, “Simulating Boolean circuits by finite splicing”, Evolutionary Computation, *IEEE proceeding* , 1999.
5. J.J. Mulawka,W.Piotr, A.Plucienniczak,“Another logical molecular NAND gate system. microneuro.” IEEE, pp. 340,1999.
6. H.Ahrabian, M.Ganjtabesh,A.Nowzari-Dalini, “DNA algorithm for an unbounded fan-in Boolean circuit” Biosystems ,pp.52-60,2005.
7. W.Liu,X.Shi,S.Zhang,X.Liu,and J.Xu,“A new DNA computing model for the NAND gate based on induced hairpin formation.” Biosystems, pp.87-92,2004.
8. M. Kadkhoda, A.Pouyan, “DNA-based simulation model for bounded fan-in Boolean circuits”, *Proceedings of the 10th WSEAS international conference on Computers*,pp. 1231-1235.2006.
9. E.Shapiro,and B.Gil, “Biotechnology: Logic goes in vitro. Nature nanotechnology”, pp. 84-85,2007.
10. C.M. Gearheart,E.C.Rouchka ,B. Arazi, “ DNA-Based Active Logic Design and Its Implications”, Journal of Emerging Trends in Computing and Information Sciences,2079-8407,2012.
11. B.S.E.Zoraida,M.Arock,B.S.M. Ronaldet,R.Ponalagusamy , “ A novel generalized design methodology and realization of Boolean operations using DNA”, Biosystems, pp.146-153,2009.