Design of Full Adder and Full Subtractor using DNA Computing

Ashish Lamaniya
Department of Computer Sc. & Engineering, SRIT, RGPV University, Jabalpur, India

Prof Brajesh Patel
Head of Computer Sc. & Engineering Department, SRIT, RGPV University, Jabalpur, India.

Abstract - DNA-based computing is at the intersection of several threads of research. The information bearing capability of DNA molecules is a cornerstone of modern theories of genetics and molecular biology. The information in a DNA molecule is contained in the sequence of nucleotide bases, which hydrogen bond in a complementary fashion to form double-stranded molecules from single-stranded oligonucleotides. In 1994, L. Adleman's proof of concept that recombinant properties of real DNA can actually use massive parallelism to solve problems appropriately encoded into single DNA strands. DNA computing is an old but a developing technology. Vast range of applications can be developed using DNA, including very complex problems like NP-complete problems. The versatile utilities inspire me to select this field as my research work. In my work I have implemented of some combinational circuits.

Keywords: Digital Circuits, Combinational Circuits, DNA, Boolean algebra, NP Problems.

I. INTRODUCTION

The manipulation of complex DNA solutions with genetic engineering tools has been proposed recently as a chemical methodology for solving instances of computationally intractable (so- called NP-complete) combinatorial problems [10]. DNA computing holds out the promise of important and significant connections between computers and living systems, as well as promising massively parallel computations. Before these promises are filled, however, important challenges related to errors and practicality has to be addressed. On the other hand, new directions toward a synthesis of molecular evolution and DNA computing might circumvent the problems that have hindered development, so far.

Boolean circuits are defined in terms of the logic gates they contain. For example, a circuit might contain binary AND and OR gates and unary NOT gates, or be entirely described by binary NAND gates. Each gate corresponds to some Boolean function that takes a fixed number of bits as input and outputs a single bit.

II. DNA COMPUTING FUNDAMENTALS

Deoxyribonucleic acid is the expansion of the abbreviation DNA which is the germ plasma of all the living types. It is a macromolecule of biology which is made up of many small nucleotides. And in that nucleotide, it is composed of a unique base out of the four varied types of it. The four bases are adenine (A), thymine (T) or guanine (G) and cytosine (C) matching to the corresponding nucleotide. The single-stranded DNA is developed with positioning of one end known as (5 prime) 5′ and the location of the other end is said to be (3 prime) 3′ [5].

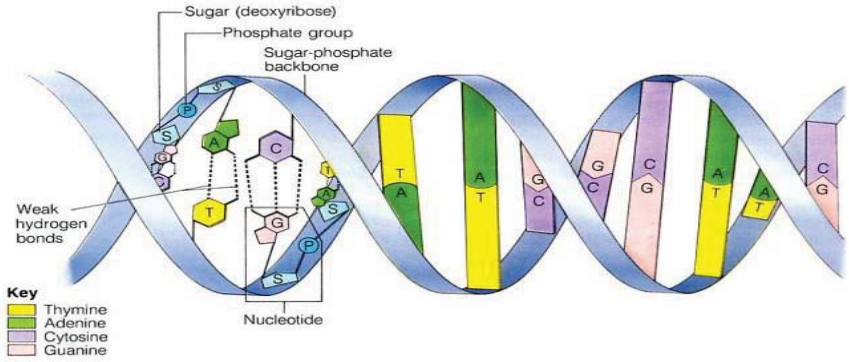


Figure 1: Basic DNA Structure

Naturally, the DNA is in the form of double helical structure or it can be said that it is a double strand molecule. The two individual complementary strands of DNA are joined together, by making a bond with the complementary (A and T or C and G) bases with the help of hydrogen bond between them for bonding. This is done to form the double-helix structure of DNA.

The biological instruction present in the strand of DNA is stored as a code which is determined by the order, or sequence, of these nitrogen bases, further used to determines the information necessary for building and maintaining an organism, similar to the way of forming words and sentences using letters of the alphabet in a certain order. For example, the sequence ATCGTT might instruct for blue eyes, while ATCGCT might instruct for brown.

2.1 DNA Computers

DNA computing performs the computations using biological molecules, rather than traditional silicon chips. DNA (deoxyribonucleic acid) molecules, the material our genes are made of, have the potential to perform calculations many times faster than the world's most powerful human-built computers. DNA might one day be integrated into a computer chip to create a so-called biochip that will push computers even faster. DNA molecules have already been harnessed to perform complex mathematical problems. While still in their infancy, DNA computers will be capable of storing billions of times more data than your personal computer [2].

DNA computers can't be found at your local electronics store yet. The technology is still in development, and didn't even exist as a concept a decade ago. In 1994, Leonard Adleman introduced the idea of using DNA to solve complex mathematical problems. Adleman, a computer scientist at the University of Southern California, came to the conclusion that DNA had computational potential after reading the book "Molecular Biology of the Gene," written by James Watson, who co-discovered the structure of DNA in 1953. In fact, DNA is very similar to a computer hard drive in how it stores permanent information about your genes.

Adleman is often called the inventor of DNA computers. His article in a 1994 issue of the journal Science outlined how to use DNA to solve a well-known mathematical problem, called the directed Hamilton Path problem, also known as the "traveling salesman" problem.
The success of the Adleman DNA computer proves that DNA can be used to calculate complex mathematical problems. However, this early DNA computer is far from challenging silicon-based computers in terms of speed. The Adleman DNA computer created a group of possible answers very quickly, but it took days for Adleman to narrow down the possibilities. Another drawback of his DNA computer is that it requires human assistance. The goal of the DNA computing field is to create a device that can work independent of human involvement [4].

DNA computer components -- logic gates and biochips -- will take years to develop into a practical, workable DNA computer. If such a computer is ever built, scientists say that it will be more compact, accurate and efficient than conventional computers.

2.2 The DNA Computation Model

To put DNA computation on a concrete framework, the article by Lila Kari breaks the process into smaller steps that can be regarded as primitive operations for the DNA computer [10]. These steps are Synthesizing, Mixing, Annealing, Amplifying, Separating, Extracting, Cutting, Lighting, Substituting Detecting and Reading.

III. BOOLEAN ALGEBRA AND CIRCUITS

In mathematics and mathematical logic, Boolean algebra is the subarea of algebra in which the values of the variables are the truth values true and false, usually denoted 1 and 0 respectively. A logic gate is an elementary building block of a digital circuit. Most logic gates have two inputs and one output. At any given moment, every terminal is in one of the two binary conditions *low* (0) or *high* (1), represented by different voltage levels. The logic state of a terminal can, and generally does, change often, as the circuit processes data. In most logic gates, the low state is approximately zero volts (0 V), while the high state is approximately five volts positive (+5 V). Some circuits may have only a few logic gates, while others, such as microprocessors, may have millions of them. There are seven basic logic gates: AND, OR, XOR, NOT, NAND, NOR, and XNOR.

3.1 Full-Adder (FA) circuit

Full adders are used to add three bits where one of them is the carry from the preceding adder. They have two outputs: the sum and the carry to the next stage. The block diagram of full-adder is shown below.

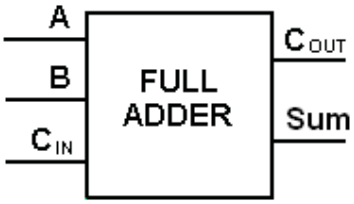


Figure 2: Block Diagram of Full Adder

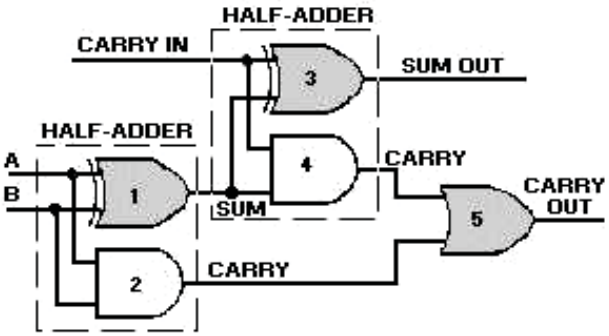
Basically full-adder is a combination of two half-adders. As shown in Figure-3 the full adder has three inputs A, B, and Cin (named as C in following equations). And it has two outputs Sum and Cout. The Boolean equations of Sum and Cout are as follows:

$$\text{Sum} = A'B'C + A'BC' + AB'C' + ABC$$

$$= A \text{ xor } B \text{ xor } C$$

$$\text{Cout} = AB + A'BC + AB'C$$

$$= AB + (A \text{ xor } B)\, C$$



A	B	CARRY IN	SUM OUT	CARRY OUT
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Figure 3: Circuit Diagram and Truth Table of Full Adder

3.2 Full Subtractor

There are full subtractors with three inputs one of which is the borrow to the preceding subtractor. The two outputs are the difference and the borrow from the succeeding unit. Figure 4 shows a block diagram of a full subtractor.

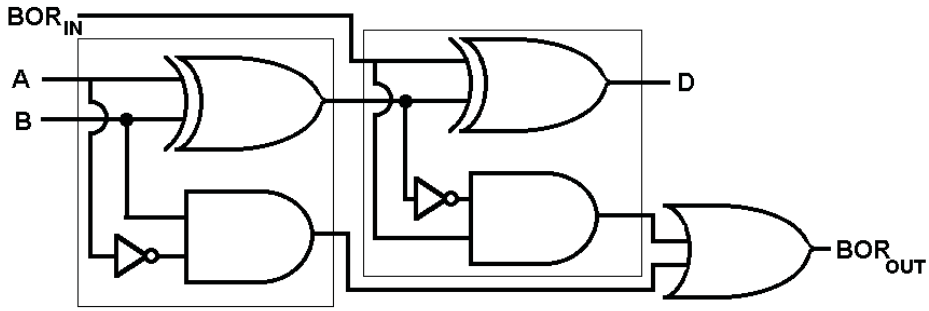
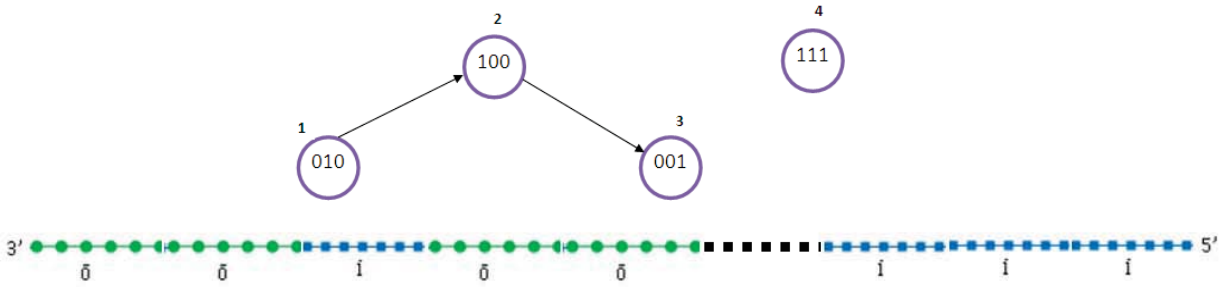


Figure 4: Circuit Diagram of Full Subtractor

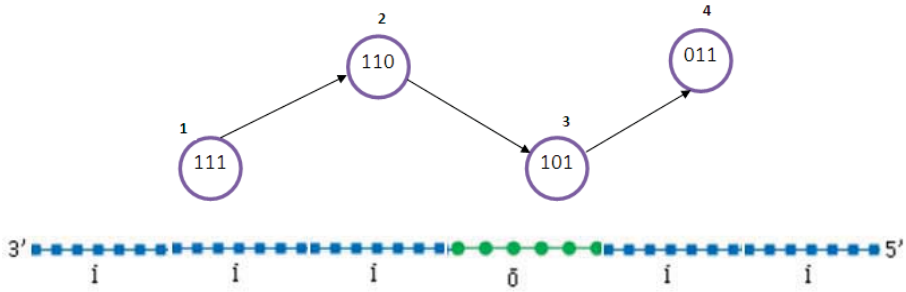
IV. 4. DNA COMPUTING SOFTWARE

The use of true DNA computers may not be possible, as there don't exist commercially at the moment. It seems perfectly reasonable to try to build software to simulate one. That was the motivating idea for this program. The purpose of this program is only to help designing algorithms for DNA computers. In following section I give the design concept of Full-adder and Full-subtractor. The system implementation of basic building blocks of Boolean circuit is shown in following section by displaying the graph representation of Boolean elements and their corresponding DNA strands.

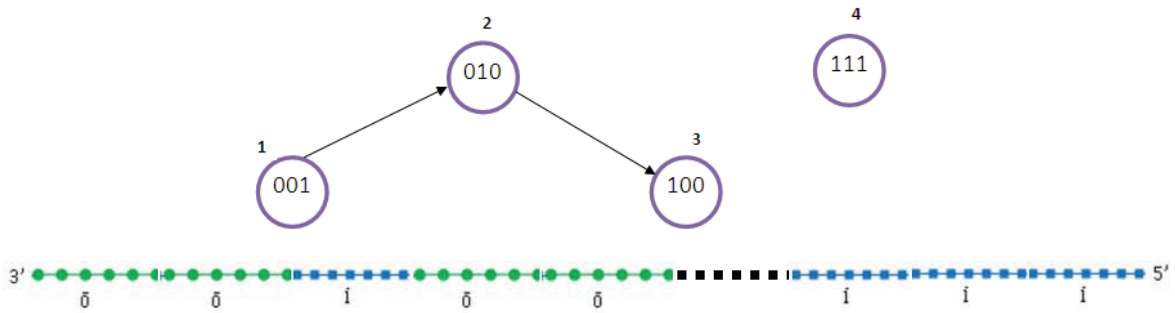
4.1 Full Adder: Graph representation and DNA strand for the Sum-Out of Full Adder is as follows:



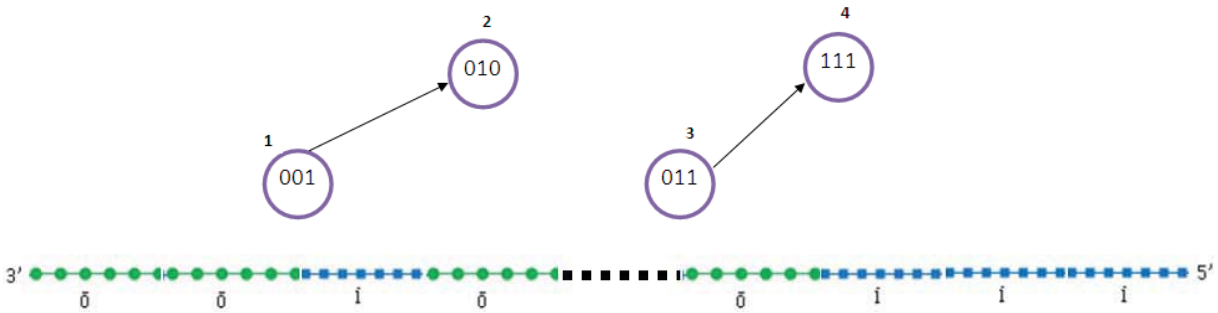
- Graph representation and DNA strand for the Carry-Out of Full Adder is as follows:



4.2 Full Subtractor: Graph representation and DNA strand for the Difference of Full Subtractor is as follows:



- Graph representation and DNA strand for the Borrow-Out of Full Subtractor is as follows:



V. CONCLUSIONS

The field of DNA computing remains alive and promising, even as new challenges emerge. Most important among these are the uncertainty, because of the DNA chemistry, in the computational results, and the exponential increase in number of DNA molecules necessary to solve problems of interesting size. Despite these issues, definite progress has been made both in quantifying errors, and in development of new protocols for more efficient and error-tolerant DNA computing. Michael Arock et al. [7] presented a theoretical model using molecular beacons. They considered the AND gate for implementation details. In this paper I implemented small Combinational circuit like Full Adder and Subtractor. This design requires proper testing in DNA computers. .

VI. ACKNOWLEDGEMENTS

I feel pleasure in conveying my profound thanks to my supervisor Prof. Brajesh Patel Head of Department Computer Science & Engineering, SRIT, Jabalpur, for his constant Support, valuable Guidance and Encouragement. I also thankful to all staff members of Computer Engg for their time to time supports. I am also very thankful to my family members for their cooperation and motivation.

REFERENCES

[1] Danny Van Noort and Laura F. Landweber,2005, ”Towards a re-programmable DNA computer”, Natural Computing , Springer.
[2] Giuditta Franco and Vincenzo Manca, 2005, “An algorithmic analysis of DNA structure”, Soft Computing, Springer-Verlag.
[3] Kashif Hame, 2011, “DNA Computation Based Approach for Enhanced Computing Power”, Department of Computer Science, The Islamia University of Bahawalpur, Bahawalpur, Pakistan, IJES, March.
[4] Lila Kari and Kalpana Mahalingam, 2009, “Watson–Crick palindromes in DNA computing”, Springer.
[5] Lila Kari, 1997,“DNA Computing: Arrival of Biological Mathematics”, Springer-Verlag New York Volume 19, Number 2.
[6] Marc Blenkiron, D. K. Arvind, and Jamie A. Davies, 2007,“Design of an irreversible DNA memory element”, Springer.
[7] Michael Arock , R.Ponalagusamy , and B.S.E.Zoraida , 2010, “An Efficient Algorithm for Constructing DNA Boolean Circuit “, International Journal of Computer Applications (0975 - 8887),Volume 1 – No. 22.
[8] Michael S. Livstone, Ron Weiss and Laura F. Landweber, 2006, “Automated design and programming of a microfluidic DNA computer”, Natural Computing, Springer.
[9] Russell Deaton, Max Garzon, John Rose, D. R. Franceschetti, 2010, “DNA Computing: A Review”, Fundamenta Informaticae 30(23), IOS Press.
[10] Shrenik Shahy, Harvard University, Cambridge, 2009, ” APPLIED MATHEMATICS CORNER DNA Computation and Algorithm Design”.
[11] Yasubumi Sakakibara, 2003, “DNA-based algorithms for learning Boolean Formulae”, Natural Computing 2: 153–171, Kluwer Academic Publisher, Japan.
[12] Z. Frank QIU and Mi Lu, Texas A&M University, 2000, “A new solution for Boolean Circuit with DNA Computer”, April 10.