Vishnu Shankar

$E = mc^2$ Publication Submission

July 1, 2014

My Intel Science Talent Search project involved studying G-Protein coupled receptors (GPCR). These receptors are specific transmembrane proteins that convert an extra-cellular signal to an intra-cellular response, thereby controlling signaling cascades in cells. They are immensely important since they control virtually every physiological response in the body. My project involved computationally predicting the structure of a GPCR for therapeutic purposes, specifically for finding medications that interact with the GPCR.

As a freshman in high school, I took both programming and biology classes. Biology fascinates me in the way it attempts to characterize and explain living things in the world. Programming interested me in its potential as an effective tool to solve real-world problems by harnessing the ability of computers to solve major problems. The synergy of biology and computing allows for a variety of applications from drug development to studying population dynamics and molecule transport. The ability to solve biological problems or explain biological phenomenon with computing powers definitely whet my interest, especially due to the immense applications from combining biology and computing.

The majority of my GPCR research was performed at California Institute of Technology from 2012 to present. Performing my research required additional understanding in biology, chemistry, and physics. From a biological standpoint, I had to learn about GPCRs, how they function, and their signaling mechanism. From a chemical standpoint, I had to understand hydrogen bonding, kinetics, free-energy. Finally, from a physical standpoint, I had to understand Hooke's Law and dielectric constants. Due to the multidisciplinary nature of the research, I had to study across many fields to gain understanding into my area of research.

To this journal, I have submitted my research in DNA computing. This research was conducted at the NASA Ames Laboratory from 2010-2011. My interest for pursuing DNA Computing is based on the limitless applications of biological and computing sciences combined. Additionally, I remember reading about the potential of DNA computing in a Scientific American article, which showed a DNA "computer" playing tic-tac-toe against a human. The Scientific American article augmented my interest in studying the applications of computing with organic molecules. My DNA computing project involved building a biological half-adder or a system that can add single bit integers based on organic molecules and evaluating the energy efficiency of that system. Aside from the scientific components, this project introduced me to how research was conducted in the real world. I explored a world of gathering exotic materials, tailoring proposals, and convincing investigators to lend their lab for a few experiments. To complete this project, I had to understand various biological techniques in the lab (i.e. Polymerase Chain Reaction, Gel Electrophoresis, blotting techniques). I had to also learn basics of computing through ideas such as logic gates, different logical operators, and how a transistor works.

To young upcoming investigators, I would like to offer two pieces of advice. First, always be engaged in the literature. Being able to discuss a given topic with another scientist using the terms of art in the field helps gain proficiency and come up with new ideas. Second, never give up on your ideas. My DNA computing project was the result of lot of determination and a strong desire to expose DNA computing's potential by comparing its energy efficiency with silicon computing equivalents.

Finally, I am blessed to have relentless support from my parents and my sister, who have constantly supported my interests in science and mathematics. Specific to my DNA computing project, I would like to thank Dr. Brad Bebout for generously lending me his lab space, Dr. Joanne Macdonald for teaching me about half adders and lending me some materials, and Dr. Ronald Birrell for his laboratory expertise.

A Naturally Efficient Computing Technique using Molecular Logic Gates with a DNA-cleaving Deoxyribozyme

Vishnu Shankar

Abstract: DNA based solution phase computing circuits for potential use as computers are analyzed and energy efficiency is calculated. Previously constructed DNA-computational systems can add the inputs of 1 and 0 through the use of "deoxyribozyme" molecules or DNA-based molecules with one RNA base and fluorescent dyes. If all inputs are present, the deoxyribozyme is cleaved at the RNA base, and the increase in fluorescence would indicate a "1." When insufficient inputs are added to the gate, the gate indicates no or minimal activity via fluorescence. Further, this analysis calculates the approximate energy required for a silicon system to add two single bit integers together and compares it with energy required for them same operation in a DNA Computing system. The calculation indicates that, in the case of simple addition, DNA Computing was over 6500 times more energy efficient. Although this is only a single case, it appears that DNA Computing has the potential to enable computing to be more energy efficient and cost-effective due to the nature of the components used to build a DNA system.

I. Purpose:

Current computational devices and techniques are based on silicon microprocessors. Computer manufacturers[1] have been increasing transistor density on computer chip microprocessors at a rate that approximates Moore's Law, which states that the amount of gates on a single chip will double every two years. Unfortunately, the application of Moore's[2] Law has been predicted to reach an end because of the physical speed and miniaturization limits of silicon microprocessors. The advantages of DNA Computing include large storage capacity and an ample a supply of DNA, making it a cheap natural resource unlike the cost of fabrication of Si-based computers. Even[3] though empirically it has been shown that DNA computation has slower cycle than a silicon system, the parallel processing capabilities of a DNA system is significant in solving NP[4]-hard

[1] Tredennick, Nick, and Brion Shimamoto. "The Inevitability of Reconfigurable Systems." *Queue* 1.7 (2003): 34. Print.

[2] Kaku, Michio. *Physics of the Future: How Science Will Shape Human Destiny and Our Daily Lives by the Year 2100*. New York: Anchor, 2012. Print.

[3] Watada, Junzo. "DNA Computing and Its Application." (n.d.): n. pag. Print.

[4] Adleman LM (1998) Computing with DNA, Scientific American, 279(2): 54-61

problems. Further motivations for studying DNA Computing[5] or the construction of molecular scale computing devices include its scale. Biological systems through superior control have been shown to solve many complex problems while avoiding the inefficiency of current von Neumann architecture[6].

II. Introduction:

DNA, the basis of all cellular machinery, is paramount to information processing in the cell. DNA is an excellent engineering material[7] for biochemical circuits because its biological nature supports technological applications in vivo, its easy chemical synthesis facilitates practical experiments in vitro, its combinatorial structure provides sufficient sequence design space, and the Watson-Crick complementary principle enables predictable molecular behavior. DNA has been used as a computing substrate[8] since the first demonstration of solving a seven-city Hamiltonian[9] path problem in 1994 and has evolved away from competing with silicon to embedding control within molecular systems. DNA Computing[10] has the potential to take computing to new levels of energy efficiency. In addition, this prototype is expected to form the basis for other macromolecular systems to engineer specific cell types and use it to sense molecular markers of disease.

III. Experimental Overview

This analysis looks at the approximate energy comparison between a molecular half adder[11] that can add inputs of 1, 0 through the regulation of inputs and produce outputs of the same type and current silicon computational models at the Assembly language level that can add inputs of 1, 0. The analysis is based in the use of logic gates to achieve Boolean calculations. When logic gates are organized into more complex circuits, they can perform more advanced functions. For instance, a YES gate takes one input and transfers the same output. Conceptually, this principle of logic gates can be translated to living organisms and DNA. One of the motivations to test energy efficiency is to confirm whether parallel computing is more efficient than sequential computing in both DNA and silicon systems. In the DNA Computing system, computing is completed in a parallel way as

[5] Maley, Carlo C. "DNA Computation: Theory, Practice, and Prospects." *Evolutionary Computation* 6.3 (1998): 201-29. Print.

[6] Henkel, Christiaan A. "Towards Evolutionary DNA Computing." *The ACM Digital Library* (n.d.): 2005. Print.

[7] Maley, Carlo C. "DNA Computation: Theory, Practice, and Prospects." *Evolutionary Computation* 6.3 (1998): 201-29. Print.

[8] Qian, Lulu, and Erik Winfree. "Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades." *Science Magazine* 332.6034 (2011): 1196-201. Print.

[9] Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. Science, 266, 1021–1024.

[10] Suruchi S., Dhiraj B., Yamuna K.,"Transforming bases to bytes: Molecular computing with DNA" , Current trend in science, 2009.

[11] Stojanovic, M. N. and Stefanovic, D. (2003) Deoxyribozyme-based half-adder. J. Am. Chem. Soc. 125, 6673–6676.

nucleotides are added simultaneously. On the other hand, silicon computing is done sequentially but can be parallelized with a suitable program. Our analysis uses a system of DNA-based logic gates, which requires two inputs and conducts the binary addition calculation. To conduct a direct comparison between DNA Computing and Silicon based computing, addition was coded in Assembly language. This allowed us to directly control the processor's actions through the use of machine code. The energy efficiency from DNA Computing derives from the fact that since all activities of the living beings are driven by DNA or a different active form of "DNA Computing" in a parallel fashion, this type of computing will be much faster than silicon computing even if the energy cost of a single operation is same in both systems. In addition, we extrapolate that since DNA is a part of every cell of living beings, the energy of DNA-based computing is lower than the energy of silicon based computing.

IV. DNA Computing Overview

The first elementary[11] step to building any computing system is the ability to use functions to perform basic operations. Addition, the basic function performed in this experiment, is conducted by using to add single bit integers producing a *half-adder*. The half adder is a key building block for a full adder, which in turn can add three bits and can be cascaded to yield serial adders for adding larger (multi-bit) integers. Consequently, the ability to construct a circuit that behaves as a half-adder is the crucial first test for any new computation system. The half-adder in this analysis consists of two separate logic gate systems.

AND gate			**XOR Gate**		
Input A	**Input B**	**Output**	**Input A**	**Input B**	**Output**
0	**0**	**0**	**0**	**0**	**0**
0	**1**	**0**	**0**	**1**	**1**
1	**0**	**0**	**1**	**0**	**1**
1	**1**	**1**	**1**	**1**	**0**

Half-Adder Logic Function

Input A	Input B	Sum	Carry Out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 1 Description: A half-adder is comprised of an XOR gate system and an AND gate. The XOR gate system or *exclusive-or* system, defaults to true if only one input is present. The AND gate shows true only if both inputs are present. This is important because in a half-adder the AND gate and XOR gate act together. Having just an XOR gate is insufficient because this would not cover the case where both of the inputs are present. Also, an AND gate does not cover the cases in addition with only one input.

V. Experimental Design

The first challenge was to design molecular logic gates[12] using DNA-based molecules as inputs, outputs, and switches similar to a silicon-based logic gate. Each molecular gate consists of 3 components including 1) the substrate or the switch[13],[14] part of the logic gate, 2) fluorescent[15] dyes to monitor the output, and 3) input molecules that bind to the oligonucleotide-binding region.

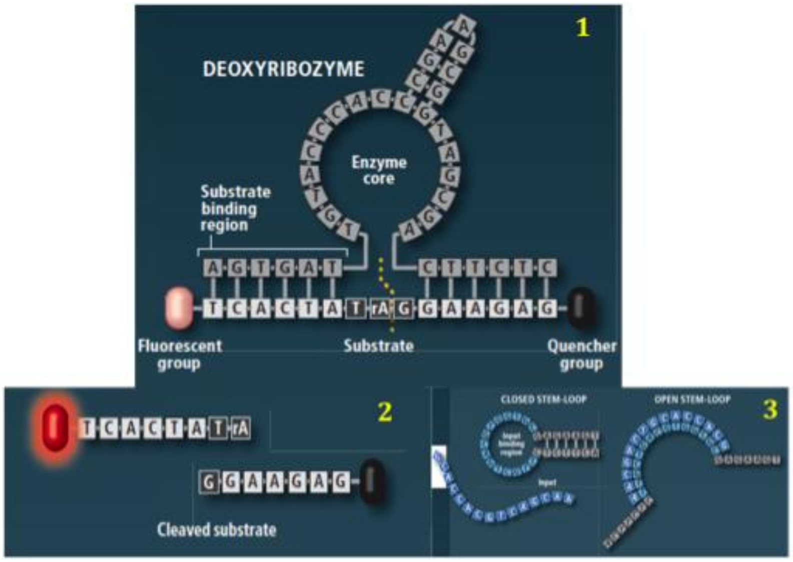


Figure 1 Description[16]: The three figures above describe the different components of a logic gate. The part marked as **enzyme core** is the switch part of the molecular logic gate. This is a nucleic acid that causes the DNA reactions. **2)** The substrate or switch part of the logic gate utilizes a deoxyribozyme, a nucleic acid based enzyme. **3)** Stem-loops can be placed in the arms of this oligonucleotides . These loops are activated with a specific sequence of nucleotides. Each loop remains closed when the nucleotide sequence complementary to the one in the loop is not mixed with the deoxyribozyme in the chemical solution. This is the inactive form of the gate. When the complementary sequence exists in the solution the loop opens and the oligonucleotides anneal causing the gate to become active.

[12] Stojanovic, Milan N., Tiffany Elizabeth Mitchell, and Darko Stefanovic. "Deoxyribozyme-Based Logic Gates." *Journal of the American Chemical Society* 124.14 (2002): 3555-561. Print.

[13] Breaker, R. R. (2002) Engineered allosteric ribozymes as biosensor components. Cur. Op. Biotech. 13, 31–39

[14] Breaker, R. R. and Joyce, G. F. (1995) A DNA enzyme with Mg2+-dependent RNA phosphoesterase activity. Chem. Biol. 21, 655–660.

[15] Tyagi, S. and Krammer, F. R. (1996) Molecular beacons: probes that fluoresce upon hybridization. Nature Biotechnol. 14, 303–309.

[16] J. MacDonald, et.al, “DNA Computers for work and play”, Scientific American, 2008.

Two Types of Logic Gates – Half Adder

Since unmodified DNA structures are not capable of causing reactions[17], DNA is configured to have enzymatic functions. Specifically, two kinds of DNA based enzymes are utilized including E6 and 8-17. The E6 enzyme has a catalytic core and an internal loop. The internal loop can be substituted by a desired sequence. An unchangeable catalytic core and a fixed internal loop form the 8-17 deoxyribozyme. When the complementary sequence exists in the solution the stem loop opens and the oligonucleotides anneal causing the gate to become active. In this case, the enzyme is a molecule that cuts the oligonucleotide substrate into shorter oligonucleotide products. An oligonucleotide is a short polymer of fifty of fewer bases. Further, the presence of two different gates was needed to produce the two different functions required for a half adder including sum and carry.

Logic Gate Mechanism

To demonstrate truly digital behavior we needed a definitive way to ascertain whether the output was a 1 or 0 according to the activity of the logic gate. To monitor the output, we labeled the outputs with fluorescent dyes. Our substrate molecule consists of a dye and a molecule for quenching the fluorescence of the molecule. In this analysis, we used the TAMRA (T) as dye and the Black-Hole 2 (BH2) as quencher. The diagram below illustrates the basis of our mechanism for determining a 1 from a 0.

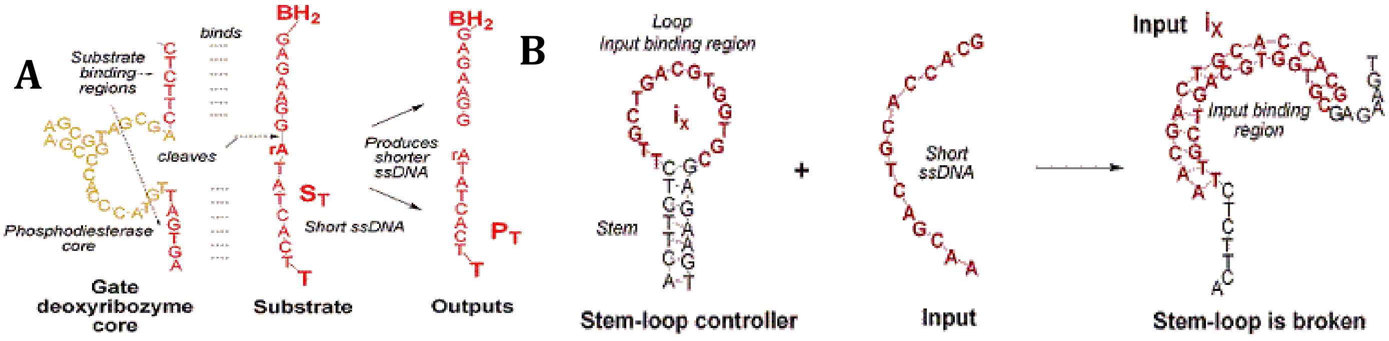


Figure 2[18]: **A)** This diagram shows the mechanism of the logic gate. Cleavage of the substrate molecule would result in a noticeable increase in fluorescence indicating a 1. If cleavage does not occur, fluorescence does not increase resulting in a 0. If the substrate molecule is separated along the RNA base, there is an increase in fluorescence because the quencher no longer absorbs the fluorescence. It is important to note that this drastic increase only occurred when all the inputs were added in the logic gate system. Without these inputs, the logic

[17] Ruiz-Perez, Maria B. "Logic Gates Made with DNA." (2002): n. pag. Print.

[18] Stojanovic, M. N. and Stefanovic, D. (2003) Deoxyribozyme-based half-adder. J. Am. Chem. Soc. 125, 6673–6676.

gate would show minimal or no activity. The input molecules specifically regulate the cleavage through the activity of the stem loop. **B)[19]:** The mechanism for a stem loop is illustrated in the diagram above. With the appropriate DNA input, the stem loop is broken.

The final step for the logic gate is to regulate the input DNA through a nucleotide sequence, which contains oligonucleotide-binding regions. If the correct input DNA (a short single-stranded oligonucleotide) is added, it will hybridize to the oligonucleotide binding region, causing the stem-loop to break apart. DNA inputs will only bind to their respective complementary sequence. This presents some practical implementation considerations. DNA input molecules would have to be very specifically designed such that, if we were to mix multiple gates together in a solution, we could have multiple inputs without cross-activation of gates.

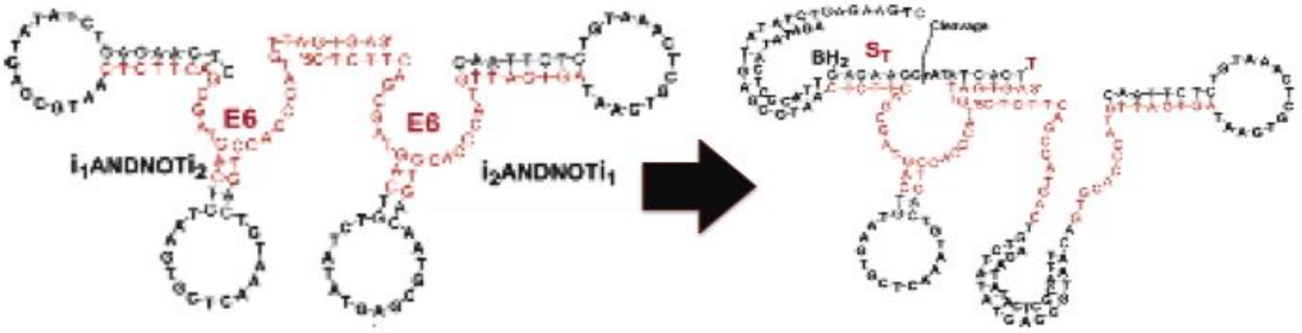


Figure 3 Description[20]: In this diagram, the enzymatic core E6 catalyzes activation of the gate. The diagram above schematically represents the change in state of the gate with the addition of a single input[21] or i1. The insertion of this gate inhibits one of the gates as seen in the 2nd part of the diagram. If both inputs were present, both gates would be inhibited.

To make a half-adder, we need to be able to construct an XOR gate system along with an AND gate, which is made by activating two stem loop structures. This is because both stem loop (*regulating)* regions would need to be activated because in order for an AND gate to evaluate to 1 or a ‘TRUE,’ both inputs are needed to activate the gate. Also, a XOR gate system can be constructed using 2 ANDNOT gates. The ANDNOT gate is activated in the presence of only one input. If the second inhibitory input is added, then the gate has no activity. By combining these ANDNOT gates together, we were able to construct an XOR gate system. It is important to

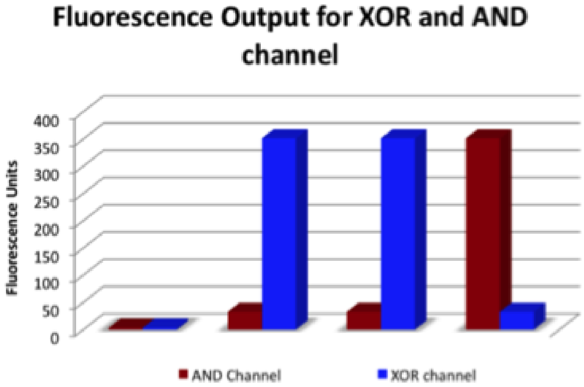
[19] Stojanovic, M. N. and Stefanovic, D. (2003) Deoxyribozyme-based half-adder. J. Am. Chem. Soc. 125, 6673–6676.

[20] Ibid.

[21] J. Macdonald, et.al, “Medium Scale Integration of Molecular Logical Gates in an Automaton”, Nanoletters, Vol.6, No.11, p.2598-2603, 2006.

note that to make other gates (i.e. *YES gate, NOT gate*), one just simply needs to change the stem loop region (*given that the inputs are changed accordingly)* because the other components of the logic gate are identical.

VI. Half-Adder Results



INPUT A	0	1	0	1
INPUT B	0	0	1	1

Figure 4 Description: The system conducted binary addition as follows. The absence of both inputs translates to a lack of activity in both the XOR system and the AND system. This remains consistent with the intended function where no inputs resulted in a 0 in the sum function. The presence of either and only 1 input activates the XOR gate marked by the increase in fluorescence. However, the AND gate system was inactive. This is consistent with **Table 1** where the presence of 1 input, translates to an increase in fluorescence in the AND gate. Finally, the presence of both inputs in the system activated the AND gate, while there was a lack of activity in the XOR system. This remains consistent with binary addition calculations, where two inputs added together results in a 0 with a carry of 1. The 0 is seen through the XOR system, while the 1 (increase in fluorescence) is seen through the AND gate.

VI. Materials and Equipment – DNA Computing

Equipment: Fluorescence Plate Reader with excitation filter 530 nm for TAMRA and Black Quencher, Black 384 Well plates, and Centrifuge

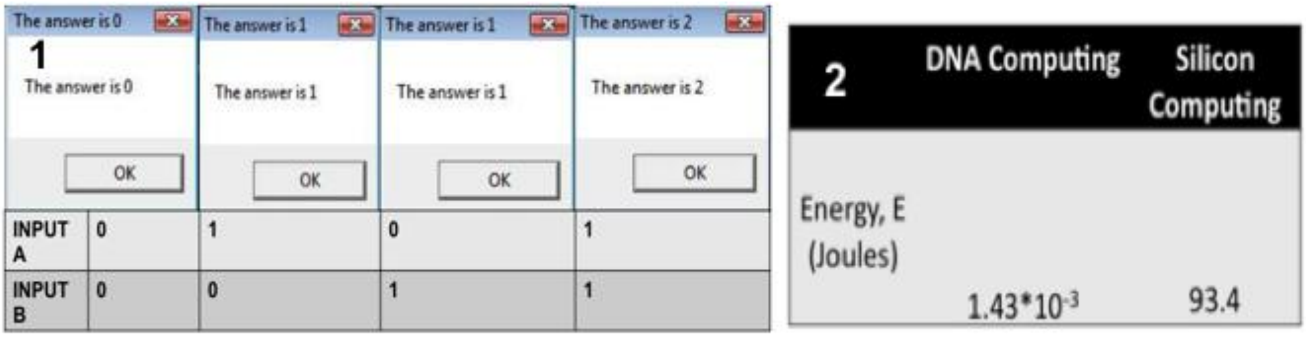
Specific Procedure[22] – DNA Computing

Prepare 500 µL of a 100 nM gate solution of the type iAANDANDNOTiB (*iA, iB)* in reaction buffer containing Zn2+. Add substrates (for E6 gate derivates) or SF (for 8-17 gate derivates) to a final concentration of 2.5 µM, and vortex.). Distribute 50 µL of solution in 8 wells of a black 384-well plate. Prepare eight 10 µL input solutions in buffer containing Zn2+. The first solution contains 10 µM of each iA, iB the second three solutions contain each of the three possible combinations of the two inputs, the third three solutions contain individual inputs, whereas the last solution is blank. Add 5 µL of each of the input oligonucleotide solutions into individual wells. Follow the fluorescence increase in all wells over 90 min with measurements every 15 min, using a multilabel fluorescence reader containing fluorescein.

[22] Macdonald, Joanne, Darko Stefanovic, and Milan N. Stojanovic. "Solution-Phase Molecular-Scale Computation With Deoxyribozyme-Based Logic Gates and Fluorescent Readouts." *Methods in Molecular Biology* 335 (n.d.): 343-63. Print.

VII. Results

To effectively compare the ability of a DNA system to conduct addition, we compared the DNA addition of a silicon computing system. To resemble the simple system in DNA Computing, we used the assembly language to code for a half-adder program as it gave us control to manually deal with the hardware of a silicon system. Assembly language is machine specific and considered a "low level" language. This means that the code and syntax is much closer to the computer's processor, memory, and I/O system. Using a specific program[23], we found the approximate energy required to add two single bit integers. This was compared to the energy required for the logic gate to fold. Then, we estimated the energy required for the computer to function and compared it with the energy required for the DNA computing system.



The answer is 0

The answer is 1

The answer is 1

The answer is 2

INPUT A	0	1	0	1
INPUT B	0	0	1	1

2	DNA Computing	Silicon Computing
Energy, E (Joules)	$1.43*10^{-3}$	93.4

Figure 5 Description: (1) The lab data remained consistent with the truth table logic of a half-adder. The diagram above demonstrates the output of both the XOR and AND channel with different variations of outputs. By using the AND gate data and the ANDNOT gate data(*consistent with predictions),* we constructed a half-adder. It can be seen from the data that with no inputs, there was no activity from the gate. With the presence of only one input, the XOR gate showed great activity while the AND gate showed little activity. When both inputs were present the XOR channel showed no activity, while the "Carry-Out" (*see Table 1)* column gets the 1. (2) We compared the energy efficiency[24] of the DNA Computing system against the Silicon system as follows: We used an externally available software to determine the energy required for the DNA to fold in the specific logic gate pattern. We compared this with the energy required for the computer to perform the given

[23] "Joulemeter - Microsoft Research." *Joulemeter - Microsoft Research*. N.p., n.d. Web. 2012. <http://research.microsoft.com/en-us/downloads/fe9e10c5-5c5b-450c-a674-daf55565f794/>.

[24] DNA nearest-neighbor thermodynamics. Proc. Natl. Acad. Sci. USA 95, 1460–, 1465

task of addition. With just these estimations, we illustrated that that DNA Computing is more energy efficient than a silicon computer by 6500 times.

VIII. Discussion – Important Distinctions between DNA and Silicon Computing Systems

This analysis reveals many critical distinctions between DNA and silicon computing systems. In our experiment, the autonomous nature of the logic gates allows the DNA molecules to “make decisions”[25] based on the respective inputs. One of the computational limitations on the logic gates included our inability to reset the gates. Although the cleaved oligonucleotides with the fluorophores cannot be re-assembled, we predict that the cost required to produce a different gate function is still many times cheaper than silicon system assembly costs.

Unlike silicon systems, which have no intermediate values between 1 and 0 because digital behavior is determined by voltage, the DNA gate system can change from inactive to active forms, where DNA systems can. Further, computing with DNA has significant advantages over silicon machines. The biggest and most obvious advantage is the parallel computational[26] abilities of DNA computing. Current computing systems based on von Neumann architecture handle instructions sequentially. In other words, a single computation must be completed before the following computation can begin. The major advantage of DNA computing involves the capacity for multiple computations to take place simultaneously. The parallel abilities of DNA systems are much faster than of silicon systems. A mix of 1,018 strands[27] of DNA could operate at 10,000 times the speed of today's advanced supercomputers. Other major advantages[28] of DNA-based computing include the immense memory capacity, its clean and cheap nature, and low power dissipation.

[25] J. MacDonald, et.al, “DNA Computers for work and play”, Scientific American, 2008.

[26] Parker, Jack. "Computing with DNA." *EMBO Reports* (2003): n. pag. Print.

[27] Parker, Jack. "Computing with DNA." *EMBO Reports* (2003): n. pag. Print.

[28] Tagore, Somnath, Saurav Bhattacharya, Md Ataul Islam, Sucharita Dey, and Md Lutful Islam. "DNA Computation: Applications and Perspectives." *Journal of Proteomics & Bioinformatics* 03.07 (2010): 234-43. Print.