A Comprehensive DNA Arithmetic Calculator[*]

Shaw-Jye Wu,[†] Yuan-Jye J. Wu,[‡] and Joe J. Shaw[§]

November 18, 1996

Abstract

Developing a generalized useful DNA computer has been one of the main interests since the inception of molecular computation. Toward that goal it is important to equip DNA with the ability to perform mathematical calculations. Here, a simple procedure that allows DNA-based bit arithmetic calculations is presented. Oligonucleotides encoding input bit numbers serve as primers for PCR to carry out the arithmetic operations. Amplified DNA fragments encoding output bit numbers are subsequently modified to assume the same format as the input. Thus, either iterative or parallel operations are permissible.

1 Introduction

After Adleman exploited DNA to solve a directed Hamiltonian path problem [1], the feasibility of using biomolecules and their experimental protocols as a means for solving computational problems moved from the realm of hypothetical to practical. This technology takes advantage of the potential parallelism of nucleic acid operations, as well as the unique answer-screening strategies that are inherent in DNA computation. These strategies are such that combinatorial problems can be solved by constructing all possible answers at once and directly picking out the correct answer. In addition to this elegant pioneer work, several other models have been proposed to solve other combinatorial problems, such as the SAT problem [4] and the traveling salesman cycle problem [5, 9]. These models unquestionably further demonstrate the potential of DNA-based computing.

However, unlike general-use electronic computers, DNA-based computers do not employ bit manipulation. In order to develop a universal DNA computer, it would be beneficial to equip DNA with a similar bit-manipulation ability. Recently, Guarnieri et al. [3] reported an algorithm

[*] This work was supported in part by Mr. Charles Holman and by the Alabama Agriculture Experiment Station.

[†] Department of Botany and Microbiology, Auburn University, Auburn, AL 36849, wushawj@mail.auburn.edu

[‡] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, jwu@mcs.anl.gov

[§] Armstrong Laboratory/AOEL, Brooks Air Force Base, TX 78235, joe.shaw@vanadium.brooks.af.mil

that uses DNA molecules to achieve binary addition, a very welcome innovation in this respect. However, that work did not address other arithmetical operations, such as multiplication. Also, because the input and output were in different formats, neither iterative nor parallel operations were allowed. Below, we present a new model allowing DNA to accomplish various computations via iterative or parallel operations.

2 Model Description and Application

We first present a DNA-based eight-bit binary addition algorithm to illustrate our basic design. We define adenine and thymine to encode the binary numbers 0 and 1. Thus, an eight-base DNA molecule with the sequence 5'ATATATAT3' denotes the number 01010101. Binary number oligonucleotides N_i, for i = 00000000 to 11111111, with this AT array (value sequence) are then prepared to represent each number. Their complementary strands are denoted by $\bar{N}_i$. As shown in Figure 1, N_i also contains a flag sequence immediately adjacent to the 5' end of the value sequence. The flag sequence is distinct from, and not complementary to, the various value sequences. Its main function is to provide an orientation denoting top and bottom strands. Moreover, there is a trinucleotide G left border and a trinucleotide C right border flanking the 5' and 3' ends of N_i respectively. Another class of oligonucleotides, the operator DNA templates, O(i,j) consisting of three N_i and two signal-boosting sequences (SB and $\overline{\text{SB}}$) is also prepared. The arrangement of components in each O(i,j) is 5'N_i-SB-$N_{(i+j)}$-$\overline{\text{SB}}$-N_j3' (Fig. 1). Note that $N_{(i+j)}$ encoding the sum of $i + j$ is physically present between N_i and N_j. The signal-boosting sequence, SB, contains three C bases at its 3' end. When attached to a number oligonucleotides, these three Cs combined with the 3 Gs of N_i will create a restriction endonuclease recognition sequence for *Sma* I (Fig. 1). The oligonucleotide $\overline{\text{SB}}$ is complementary to the sequence of SB. Both value and signal-boosting sequences are unique and not complementary to each other, nor to the flag sequence.

Once the essential DNA molecules are ready, all possible combinations of O(i,j) are combined in a single tube to create an answer pool. When addition event $i + j$ is to be executed, one first assigns either i or j as the adding number and the other as the added number, e.g., i = adding and j = added. Oligonucleotide N_i for adding number and $\bar{N}_j$ for added number are then used as input primers to perform the polymerase chain reaction (PCR). In every case, two species of DNA fragments will be amplified. One is the desired product: it is full length O(i,j), and contains $N_{(i+j)}$, the correct sum. The other species, on the other hand, is a mixture of shorter DNA fragments derived from operator templates responsible for addition events i + (j-i) and (i-j) + j. As shown in Figure 2, the sum for event i + (j-i) is equal to j. These side products, however, are not a concern for reasons explained below.

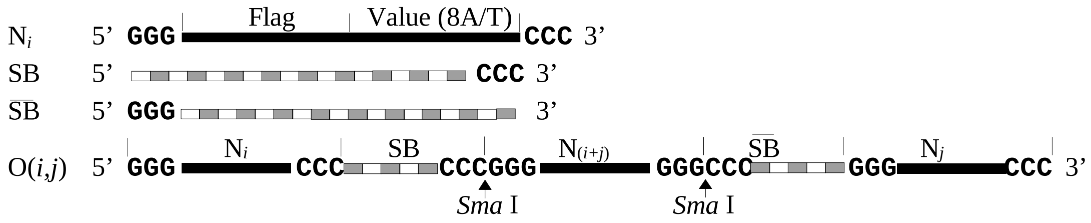

Figure 1: DNA presenting eight-bit binary numbers and the composition of operator DNA template for addition operation. Oligonucleotide N_i containing a flag sequence and a value sequence of an eight-base A/T array is used to represent an eight-bit binary number. Bases A and T in the value sequence denote binary values 0 and 1, respectively. The signal-boosting sequence (SB) contains three C bases at its 3' end. In the addition operator DNA template O*(i,j)*, these three C bases combined with the three G bases from N_i generate a recognition sequence for the endonuclease *Sma* I (arrows point to cleavage sites). $\overline{\text{SB}}$ is the complement of SB. DNA sequences of the flag, value, and signal-boosting sequences are unique and not complementary to each other.

A second round of PCR is performed on the newly amplified O(*i*,*j*) using biotinylated SB as primer. There are three purposes for this step. First of all, it discriminate the side products from the desired one. The reason is that, among first round PCR products, only the full-length O(*i*,*j*) contains two SB sequences. Thus, only the $N_{(i+j)}$ will be amplified. Second, biotinylation permits quick readout of the product by chromogenic or fluorescent detection: all values of N_i can be pre-blotted to a membrane and probed with the newly amplified SB-containing DNA fragments [6]. Third, this amplification prepares data for storage, transfer, and later retrieval.

$N_{(i+j)}$ can also be released from the signal-boosting sequences by *Sma* I digestion. Single-stranded DNA can then be affinity purified by recognizing the flag sequence. Depending upon position assignment (adding or added), $N_{(i+j)}$ or $\overline{N}_{(i+j)}$ is then ready to serve as input for the next round of addition (Fig. 2).

Although we have described a binary addition operation, the model we have presented is not limited in such scope. Several modifications can effectively enhance its versatility:

(i) Simply by substituting the value of product [$N_{(i\times j)}$] for the value of sum [$N_{(i+j)}$] in the operator DNA templates, the same protocol can carry out multiplication. Deduced accordingly, other arithmetical operations as well as comparison operations are also feasible. Since the input and output have the same molecular format, data generated from different calculations are fully compatible.

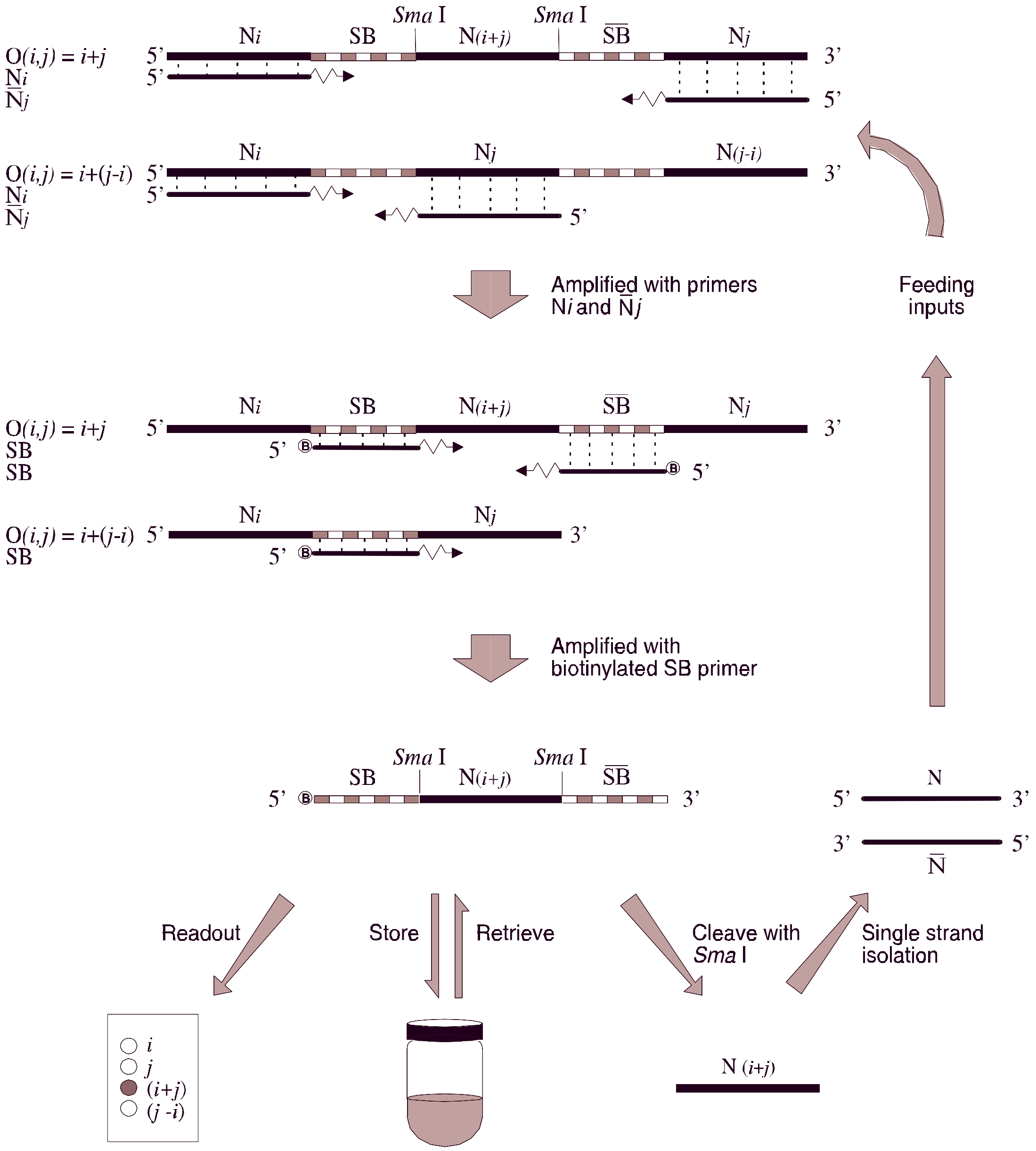

Figure 2: Illustration of addition operation as an example of DNA-based arithmetic calculations. All PCR amplified fragments are double stranded, but only a single strand is shown. Vertical dotted lines represent the positions for primer extension.

(ii) In the example presented, in order to prevent cleavage by *Sma* I, only bases A and T are allowed in the value sequences. Both flag sequences and value sequences, however, can be designed as a pentamer with one base reserved for base A or T, e.g., 5'(A/T)NNNN3'. Such a layout provides the freedom to assign distinct values to all four bases. If practiced, this modification will essentially convert the binary system into a quaternary system. A single pentamer would signify a four-bit quaternary number, while an eight-bit quaternary number would be represented by aligning two pentamers into a decamer. An instant benefit from this adjustment is the reduction of required DNA mass. More variability in the flag sequences, on the other hand, could be used to increase parallel operation power.

(iii) A cohesive-end restriction sequence can be interposed between the value sequences and the flag sequences.This type of construction permits value sequences to attach to and detach from their flag sequences freely. Consequently, number oligonucleotides (N_i) with different flags can be unified, and data-parallel operation is allowed.

To better understand how these modifications facilitate the operation, we show the application of this enhanced model to solve an instance of vector inner product problem. Assume two vectors with n components

$$v_1 = (a_1, a_2, a_3, ..., a_n) \qquad (1)$$

$$v_2 = (b_1, b_2, b_3, ..., b_n). \qquad (2)$$

The inner product of these two vectors is

$$(a_1 \times b_1) + (a_2 \times b_2) + (a_3 \times b_3) + \cdots + (a_n \times b_n). \qquad (3)$$

Suppose there is no boundary for DNA to encode the value of each component. N_i, as described before but with a restriction sequence for *Apa* I (5'GGGCCC3') inserted between flag and value sequences, is used to represent numbers (Fig. 3A). Flag sequences in this case, however, are variable. For each $k = 1, ..., n$, value sequences for a_k and b_k are tagged by the same flag sequence. Again, the various value, flag, and signal-boosting sequences are unique and not complementary to each other.

The first step of the calculation is to execute all multiplication operations at once. All pairs of N_i and $\overline{N}_j$, for $i = a_k$ and $j = b_k$, are mixed together with their flag-unique operator DNA templates for multiplication in a single reaction tube. Double-stranded DNA fragments encoding the products are then obtained concurrently by two rounds of PCR followed by *Sma* I digestion. Both strands of $N_{(i\times j)}$ and $\overline{N}_{(i\times j)}$ are affinity purified and jointly collected and allowed to resume a

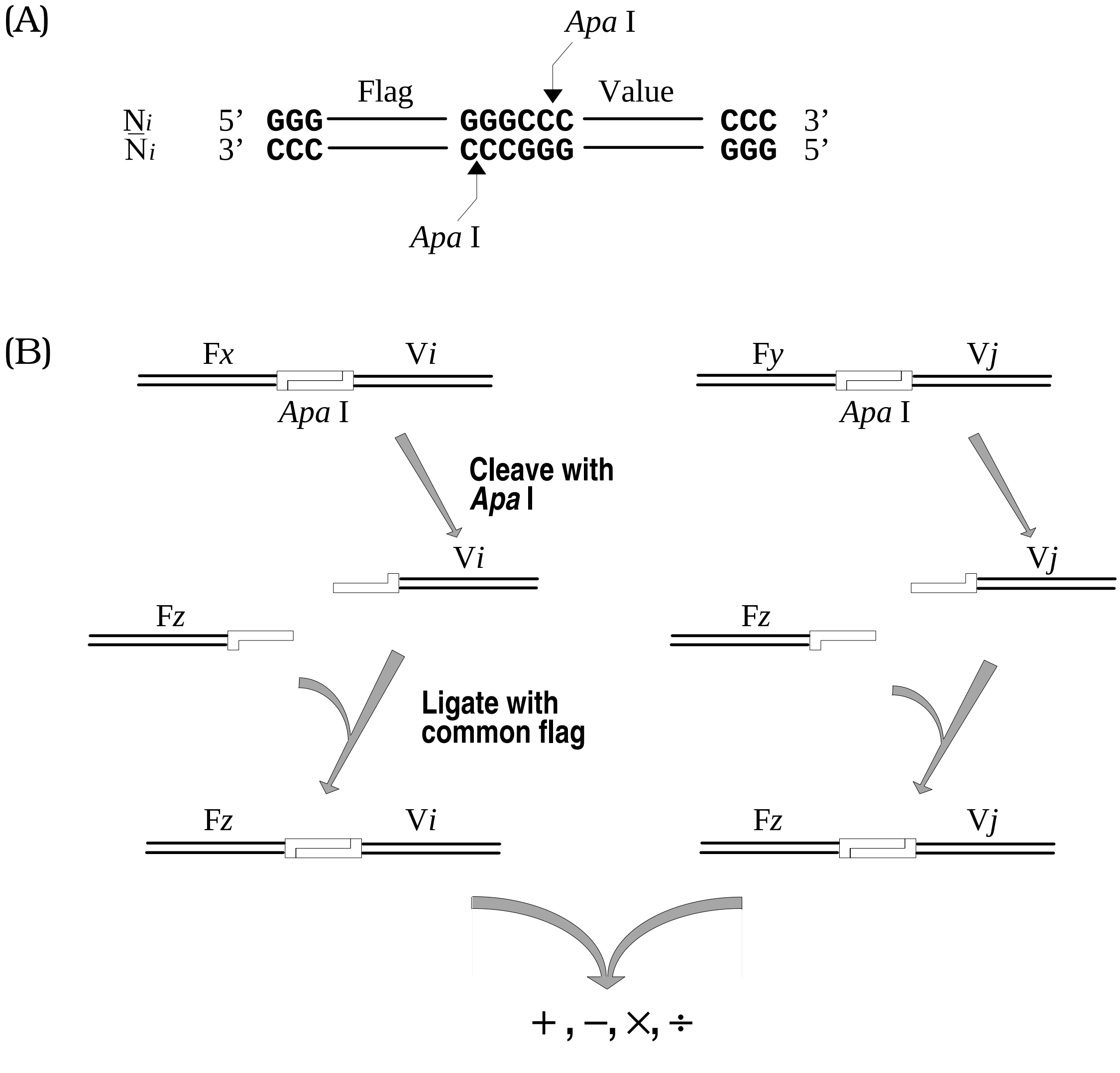

Figure 3: Unification of number oligonucleotides. (A) Modified number oligonucleotide with a restriction endonuclease sequence inserted between flag and value sequences. Shown is the sequence for *Apa* I. (B) Number oligonucleotides with different flag sequences unified by cutting out their varied flags and reattaching a common one in order to allow continual amplification by a single pair of primers.

double-stranded formation. The second step of the calculation is a chain of addition operations. For each pair of products to be added, value sequences are released from their particular flag sequences by *Apa* I digestion. A new and specific flag sequence is subsequently religated to such pair of value sequences (Fig. 3B). Once every pair of value sequences shares a specific flag sequence, a parallel addition operation is executed by the principles described previously. In this instance the operator DNA templates are now employed for addition rather than for multiplication. DNA fragments encoding the sums of this addition operation will again be subjected to the reassignment of flag sequences for another round of addition. This addition relay is continuously performed until the final answer is obtained.

3 Discussion

DNA computing is still in its infancy. Less than three years have passed since it was first demonstrated. Several fundamental issues remain to be solved before its potential and limitations are clear [2, 5, 7, 8]. The main contribution of this article, therefore, as an approach to the goal of developing a generally useful DNA computer, is in describing a model to use DNA molecules to achieve bit arithmetic calculations. The experimental techniques involved in the presented algorithm are fairly simple, disciplinarily mature, and routinely performed in most molecular biology labs. Other distinct features of this model include data storage and retrieving abilities and iterative and parallel operation capabilities.

Another consideration with DNA computing is the antagonism between performance and DNA mass. Biological reactions usually take minutes or hours to complete. Hence, the power of DNA computing resides mainly in its massively parallel capabilities. However, gaining very great parallel power suggests that a large quantity of DNA is required. This phenomenon is often seen in DNA algorithms proposed for solving combinatorial problems [5, 8]. A similar situation faced by the present model is that the amount of DNA required for operator templates rises exponentially with the number of digits *n*. Nonetheless, applying a one-base-one-digit strategy has neutralized this impact to some extent. Estimating conservatively, for the eight-bit binary addition introduced at the beginning of this article, if the flag sequence is composed of 8 bases and the signal-boosting sequence contains 17 bases, the length for an individual double-stranded operator DNA template is 100 bp. The amount of DNA required to construct an answer pool for this eight-bit binary addition is therefore about 6.6 x 10^6 bp (2^8 adding x 2^8 added x 100 bp), which is quite feasible for current PCR methods to handle. Using this figure, the human genome (6 x 10^9 bp from 46 chromosomes) could encode enough operator templates to simultaneously execute 900 eight-bit binary arithmetic operations. Furthermore, because this present model does not necessitate all template DNA to be confined in a single reaction vial, execution of 8.6 x 10^4 operations simultaneously in a 96-well thermalcycler is plausible (900 x 96).

The model presented here is definitely not the final version of a DNA-based arithmetic calculator. Further improvement in template design is desirable. However, this model does allow one to envision the possibility of directly using input as templates to generate corresponding output. Hybridization stringency is also another consideration. Moreover, due to the fact that DNA computes in a way very different from that of electronic computers, DNA-based calculators may not necessary emulate conventional computers, and to this end we have described a possible quaternary system.

References

[1] L. M. Adleman, Molecular computation of solutions to combinational problems. *Science*, 266:1021-1024, 1994.

[2] D. K. Gifford, On the path to computation with DNA. *Science*, 266: 993-994, 1994.

[3] F. Guarnieri, M. Fliss, and C. Bancroft, Making DNA add. *Science*, 273:220-223, 1996.

[4] R. J. Lipton, DNA solution of hard computational problems. *Science*, 268:542-545, 1995.

[5] Y.-M. D. Lo, K. F. C. Yiu, and S. L. Wong, On the potential of molecular computing. *Science*, 268:481-482, 1995.

[6] E. Pennisi, Chipping away at the human genome. *Science*, 272:1737, 1996.

[7] R. Pool, A boom in plans for DNA computing. *Science*, 268:498-499, 1995.

[8] W. P. C. Stemmer, The evolution of molecular computation. Science, 270:1510, 1995.

[9] S.-J. Wu, Y.-J. J. Wu, and J. J. Shaw, DNA solutions to searching for the Hamiltonian cycle and the traveling-salesman cycle. Preprint MCS-P610-0996, Mathematics and Computer Science Division, Argonne National Laboratory, October 1996.