DNA-strand molecular beacon optical processor

Qabeela Q. Thabit a,*, Alaa A. Al-Saffar b

a Department of Electrical Engineering, University of Basrah, Basrah, Iraq
b Department of Electrical Power Engineering Techniques, Southern Technical University, Basrah, Iraq

ARTICLE INFO

Keywords:
Electrical engineering
Binary modified signed digit (BMSD)
Pipeline method
Conventional method
Three-step algorithm
DNA-Strand
DNA computing
Molecular beacon MB

ABSTRACT

Due to the characteristics of the newly developed DNA computing, many researchers are interested in this specialty. One advantage of DNA is Deoxyribonucleic acid is that it has ability to resolve a Boolean circuit with various types of gates at the same time in a single level. Most of the prior models suffered from the limitations that each level of the circuit requests the gates to be of some kind. The model proposed in this work increases parallelism and reduces human intervention to a tremendous extent. When level-wise simulation is executed, the simulation for each model shows the decrease in the number of nitrogen bases used, which leads to the processing of the largest number of data with the ability to increase the length of a word, in addition to the adoption of the parallel principle of implementation. The model is designed on a mechanism which includes adder and multiplier.

1. Introduction

Traditional silicon computation has limitations such as it consumes much more power, fewer circuit dimensions, clock frequency, and heat loss as compared to the computing system based on Deoxyribonucleic Acid (DNA) [1, 2]. In other word, today the world needs high information density, operations in parallel and speed of processing devices. This is characteristically available in DNA computing or molecular computing. Therefore, DNA computing or molecular computing is preferable to using silicon computer technology. Molecular computing which uses either computerization DNA or biology computing has many benefits compared to the conventional technology such as [3, 4, 5, 6, 7]: DNA computing provides less energy consumption than silicon computer, DNA computers are faster and smaller than any computer builds until now and DNA computers provide the possibility of massively parallel processing and are considered nano-technology. DNA computing is intertwined with nanotechnology and Seeman was first introduced self-assembled nanostructures with DNA as constructing blocks [8]. Method of computation introduced in a different way by using DNA origami as the base for connection hairpins to obtain the circuit of DNA molecule strands in a spatially localized order where the logic molecule gates are located at the special positions as like in electric circuits, solving important difficulties in sketching spatially localized large-scale. DNA circuit constructed of four hairclips performs by a stochastic way and seldom obtained a reverse result [9]. One of the applications of modern DNA nanotechnology, which has taken a broad scope of research, is “DNA strand displacement”, “a common strategy for performing a wide range of nanoscale computing [10]. Process and applications to use recombining DNA in the design computer-aided design presented by Jain [11]. The principle of self-assembly of DNA and the theory of resonance energy transfer was integrated into the implementation of digital logic gates [12, 13, 14]. Over the last few decades, researchers have been looking for a powerful material to store data in large quantities in tune with developments and needs. The choice was DNA because it is small in size, high density (just 1 g of dry DNA can store about 455 Exabyte of data), and durable material [15, 16]. The DNA is kept for long periods of time up to centuries as long as it is provided storage conditions of dryness, darkness, and cooling [17, 18, 19]. The principles and possibilities of design are carried out through simulation to identify the new method effectiveness of the optical three-step BMSD addition operation in a manner depending on properties of DNA features [20, 21]. The organization of paper will be as follow: Section 2 presents basic information about DNA and then describes the related work and the applied three-step algorithm. Section 3 propose the theory which is used in this paper for adding two BMSD numbers. Section 4 proposes optical simulation adding two numbers, each one has 7-bits in the three-step algorithm of the addition operation. Section 5 introduces methods of the execution the design and shows the results of the performance simulation, Section 6 introduces an example of multiplication theory and simulation. Finally, a set of conclusions was formally presented.
2. Background

2.1. Fundamental concepts of DNA

DNA "deoxyribonucleic acid" is a string of double-stranded that consists of four "nucleotides". The four nucleotides of DNA which are called bases are "Adenine (A), Guanine (G), Cytosine (C), and Thymine (T)". DNA functions are: "proteins production coding", and "self-replication". Each deoxyribonucleotide has three main components, the first is five carbon atoms which are connected to a group of hydroxyl (OH).

| Step one T and W Transformation. |
|----------------------------------|
| X     | Y     | $T_{1,3}$ | $W_i$ |
|-------|-------|-----------|-------|
| −1    | −1    | −1        | 0     |
| −1    | 0     | −1        | 1     |
| −1    | 1     | 0         | 0     |
| 0     | −1    | −1        | 1     |
| 0     | 0     | 0         | 0     |
| 0     | 1     | 1         | −1    |
| 1     | −1    | 0         | 0     |
| 1     | 0     | 1         | −1    |
| 1     | 1     | 1         | 0     |

Table 1
The second is a group of phosphate while the third is base of nitrogenous. The chemical structure of DNA depends on a specific bond of two stringy sequences of bases. The bond keeps track of a complementarity property: Adenine attaches chemically with Thymine (A = T) and vice versa (T = A) by two Hydrogen bonds, whereas, Cytosine attaches chemically with Guanine (C=G) and vice versa (G=C) by three Hydrogen bonds. This is known as Watson-Crick complementarity". The structure of double-stranded DNA is shown in Fig. 1 [22].

The four nucleotides " Adenine (A), Guanine (G), Cytosine (C), and Thymine (T) form a strand of DNA". The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

2.2. Molecular beacon principle and applications

Molecular Beacons are single-stranded, fluorophore-labeled nucleic acid probes, able of creating a fluorescent light signal in the presence of target but are dark in the absence of target [24]. The procedure of design has to be supplied in the formation of molecular beacons (MBs). MB is target but are dark in the absence of target [24]. The procedure of design has to be supplied in the formation of molecular beacons (MBs). MB is target but are dark in the absence of target [24]. The procedure of design has to be supplied in the formation of molecular beacons (MBs). MB is target but are dark in the absence of target [24]. The procedure of design has to be supplied in the formation of molecular beacons (MBs). MB is target but are dark in the absence of target [24]. The procedure of design has to be supplied in the formation of molecular beacons (MBs). MB is target but are dark in the absence of target [24].

2.2.3. Three-step algorithm

A binary modified signed-digit number system appears in a class of number representation system which is called a binary modified signed-digit BMSD representation. During the operation of addition and subtraction in digital computers, BMSD representation boundary loads generation to one location to the left. Load generation chains are discarded by the use of operand redundant representations [33]. The representation of a traditional number with integer-radix (r > 1), each digit is permitted to suppose precisely r values (0, 1, ←, r-1). The aim of BMSD representation is to permit addition and subtraction of two-number in which no serial generation is desired over adder, that is, the time interval of the process free of the length of operands and its equal to the time wanted for the addition or subtraction of two digits. The decimal number can be realized in BMSD number style as in Eq. (1) [34]:

\[ D = \sum_{r=0}^{r-1} X_r r^i \]

where:-

\[ D \] - The decimal number.
\[ X_r \] - The r-th digit of BMSD number, \( X_r \in \{-\alpha, \ldots, -1, 0, 1, \ldots, \alpha\}, \alpha \leq r - 1. \]

\[ \alpha \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ \phi \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ W_i \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ T \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ S \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ i \] - The i-th digit of BMSD number, \( X_i \in \{-\alpha, \ldots, -1, 0, 1, \ldots, \alpha\}, \alpha \leq r - 1. \]

\[ i \] - The i-th digit of BMSD number, \( X_i \in \{-\alpha, \ldots, -1, 0, 1, \ldots, \alpha\}, \alpha \leq r - 1. \]

\[ \alpha \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ \phi \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ W_i \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ T \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ S \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ i \] - The i-th digit of BMSD number, \( X_i \in \{-\alpha, \ldots, -1, 0, 1, \ldots, \alpha\}, \alpha \leq r - 1. \]

\[ \alpha \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ \phi \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ W_i \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ T \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ S \] - The polarity of each DNA strand is determined by its two distinct ends: the 3' and, 5' end. The double helix an anti-parallel bonding of two complementary strands where the two strands are of adverse polarity [1, 23].

\[ i \] - The i-th digit of BMSD number, \( X_i \in \{-\alpha, \ldots, -1, 0, 1, \ldots, \alpha\}, \alpha \leq r - 1. \]
Table 3
Sequence assignment of variable.

| Variable | Variable Sequence 3’−5’ | Complement of Variable Sequence 5’−3’ |
|---------|-------------------------|--------------------------------------|
| X⁻¹     | TTTTT                   | AAAAA                                |
| X¹      | AAAAA                   | TTTTT                                |
| Y⁻¹     | CCCCC                   | GGGGG                                |
| Y¹      | GGGGG                   | CCCCC                                |
| X₀ or Y₀| AGCTC                   | TGCAG                                |

Table 4
T-transformation.

| T-transformation Sequences               |
|------------------------------------------|
| 3’X⁻¹Y⁻¹X₀⁻¹Y₀⁻¹X¹⁻¹Y¹⁻¹X¹⁻¹Y¹⁻¹      |

Table 5
W-transformation.

| W-transformation Sequences               |
|------------------------------------------|
| 3’X⁻¹(Y₀⁻¹ or X₀⁻¹)X¹⁻¹(Y₀⁻¹ or X₀⁻¹)Y¹⁻¹X¹⁻¹Y¹⁻¹ |

Table 6
NOT-gate.

| NOT-Transformation Sequence               |
|------------------------------------------|
| 3’X⁻¹                                      |

Table 7
T-transformation.

| T'-transformation                         |
|------------------------------------------|
| 3’X⁻¹Y⁻¹X¹⁻¹Y¹⁻¹                          |

Table 8
W'-transformation.

| Step Two: W'-transformation               |
|------------------------------------------|
| 3’X⁻¹(Y₀⁻¹ or X₀⁻¹)X¹⁻¹(Y₀⁻¹ or X₀⁻¹)Y¹⁻¹ |

Table 9
Input strand sequences.

| Inputs | Inputs Symbol | Input derived |
|--------|---------------|---------------|
| X      | Y             | X Y           |
| -1     | -1            | 5’ CALFluo5 S: AAAAA.GGGGG 5 BHQ-13’ |
| -1     | 0             | 5’ CALFluo5 S: AAAAA.TGCAG 5 BHQ-13’ |
| -1     | 1             | 5’ S:AAAAA.CCCC.S BHQ-13’ |
| 0      | -1            | 5’ CALFluo5 S: TGCAG.GGGGG 5 BHQ-13’ |
| 0      | 0             | 5’ XXXS: TGCAG.TGCAG 5 BHQ-13’ |
| 0      | 1             | 5’ S:AAAAA.CCCC.S BHQ-13’ |
| 1      | -1            | 5’ S:TTTTT.GGGGG 5 BHQ-13’ |
| 1      | 0             | 5’ S: TGCAG.TGCAG 5 BHQ-13’ |
| 1      | 1             | 5’ S: TTTTT.CCCC.S BHQ-13’ |

Table 10
NOT-gate input strand sequence.

| Inputs | Inputs Symbol | Input derived (MB) |
|--------|---------------|--------------------|
| X      | X             |                    |
| -1     | TTTTT         | 5’ S: FAM S:AAAAA 5 BHQ-13’ |
| 0      | TTTTT         | 5’ S: TGCAG 5 BHQ-13’ |
| 1      | TTTTT         | 5’ S: CALFluo5 S:TTTTT 5 BHQ-13’ |

3. Theory

This method aims to provide a design that relies on the implementation of an algorithm, in general, consisting of the following stages [25, 26, 37]: The main difference in applying this method with traditional design is applying the method to signed digit number system and this design depending on the cases which have the active output {-1 or +1}, in another word, not discuss the cases which have output (0). Three values for each Boolean variable are {-1, 0, 1} encoded in the unique form of DNA sequence. This design consists of the following three stages:-

3.1. Stage one: sequence assignment to variable

Three values for each Boolean variable are {-1, 0, 1}, encoded in the unique form of DNA sequence, i.e. unique oligonucleotide sequence is specified to each variable, indicating to its value except the (0)) whose code is shared by both variables as Table 3.

3.2. Stage two: gate strand design

Wipe every one of the rows in the truth table, here, a gate strand is constructed by taking only cases of inputs which result in outputs {-1 or +1}, in another word, otherwise the inputs are ignored if they give output equal (0). All the selection strands are sutured into a single strand to building gate as follows:

3.2.1. Step one (T-transformation)

According to Table 1, scan all rows, the first one results in active output is row 1, we take the two inputs (-1, -1) and store in array, then the subsequent row (row2) with Ti = -1 is deemed and its X value (-1) and Y value (0) and added to array which becomes (-1, -1, -1). Eventually, the following row4 with Ti = -1 is considered and its X value (0) is compared with the last value put in the array (0). Therefore, we take Y
value (-1) of the row4 to the array and the array becomes (-1,-1,-1, 0,-1).
After we finish the discussion of the inputs for a specific active output -1,
the same process from the bottom table for the cases which have output 
(1), While the other combinations which the result the output equal to 
0 are ignored. After completion, we get the array in the sequence (-1,-1,-
1,0,-1,1,1,1,0,1). Therefore, we need 10 DNA-strand (50nt) for achieving
T-transformation as compared with 18 DNA-strand (108nt) for the 
traditional method [27, 38] as shown in Table 4.
By converting each variable to its own DNA code, the next sequence is
3'TTTTT.CCCCC:TTTTT.ACGTC.CCCCC: AAAAA.ACGTC.GGGG-
G:AAAAA.GGGGG -5'

3.2.2. W-transformation
Scanning all rows, the first one results in active output Wi is row 2, we
take the two inputs (-1, 0) and save in array, then the following row4
with Wi = 1 is considered and its X value (0) is compared with the last
value saved in the array. As the values are the same, Y (= -1) is stored in
the array and the array becomes (-1,0,-1), eventually. After we finish the
discussion of the inputs for a specific active output (1), return the same
process from the bottom table for the cases which have output (-1), while

Table 11
Cycle one results.

| Operation | T   | W           |
|-----------|-----|-------------|
| (X1 + Y1) | 1-11-1-11101101100 | 000011-1000-100-100 |
| (X2 + Y2) | 11-1-111-111111000 | 0-1-111-10110-1-100 |
| (X3 + Y2) | 11110101111110000 | 0-1-1000-1-1-101-100 |

Table 12
Cycle two results.

| Operation | T   | W           |
|-----------|-----|-------------|
| (T1 + W1) | 0000100000001000000 | 1-11-10001010101000 |
| (T2 + W1) | 0-101-1001000100000 | 100000-1001010-100 |
| (T3 + W1) | 0000000000000000000 | 100101000010-1000 |

Table 13
Cycle three results.

| Operation | T (Final Sum) |
|-----------|---------------|
| (T1 + W1) | 1-11-110010101000 |
| (T2 + W2) | 1010-11101010-100 |
| (T3 + W3) | 100101000010-1000 |
the other combinations results output are equal to 0 are ignored. After completion we get the array in the sequence (-1, 0,-1,1, 0,1). Therefore, we need 6 DNA-strand (30 nt) to represent the achieving W-transformation as compared with 18 DNA-strand (108nt) for the traditional method [26, 37] as shown in Table 5.

By converting each variable to its own DNA code, the next sequence is represented the W-transformation

3’- TTTTT. ACGTC.CCCCC: AAAAA. ACGTC.GGGGG -5'

3.2.3. NOT-transformation

Enter all outputs which result from W-transformation to NOT-gate as shown in the following Table 6.

By converting each variable to its own DNA code, we need 2 DNA-strands (10nt) compared with 3 DNA-strand (18nt) for the traditional method [25, 26, 37] as shown in the next sequence is represented the NOT-transformation.

3’-AAAAA: TTTTT-5'

3.2.4. Step Two(T’-transformation)

According to Table 2, the step two addition output of T’-transformation is equal to 1 for the following combination {(1 + 1)} and equal to -1 for the following combination {(1-1-1)}. These combinations are represented by their sequences, but other combinations which result in the output equal to 0 are ignored. Therefore, we need 4 DNA-strand (20 nt) to represent the achieving T’-transformation as compared with 18 DNA-strand (108nt) for the traditional method as shown in Table 7.

By converting each variable to its own DNA code, the next sequence is represented the T’-transformation

3’- TTTTT.CCCCC: AAAAA.GGGGG-5’

3.2.5. W-transformation

According to Table 2, Scan all rows and the first one results active output W’i is (1) in row 2, we take the two inputs (-1, 0) and save in the array, then the following row4 with W’i = -1 is deemed and its X value (0) is compared with the last value saved in the array. As the values are the same, therefore, Y (=1) is put in the array and the array will become (-1,0,-1). Eventually, after we finish the discussion of the inputs for a specific active output (-1), return the same process from the bottom table for the cases which have output (1), While the other combinations in which the results the output are equal to 0 are ignored. After completion we get the array in the sequence (-1, 0,-1,1, 0,1). Therefore, we need 6 DNA-strand (30 nt) to represent the achieving W’-transformation as compared with 18 DNA-strand (108nt) for the traditional method [26, 37] as shown in Table 8.

By converting each variable to its own DNA code, the next sequence is represented the W’-transformation.

3’- TTTTT. ACGTC.CCCCC: AAAAA. ACGTC.GGGGG -5'

3.2.6. Step three (T-transformation)

Step three includ one transformation, the T-transformation which represents final sum results, in this step T-transformation of the first step is used, also we need 10 DNA-strand (50 nt) to represent the achieving T-transformation compared with 18 DNA-strand (108nt) for the traditional method [26, 37].

3.3. Stage three: input strands design

Inputs for all gate strands are supplied in the molecular beacon MB form. This stage involves the design of inputs, where each case of input...
has its own molecular beacon. MB, in actually, a DNA hairpin form with fluorophore and quencher in its two finishes. In the case of a link between the MB and the gate, the fluorescence end is separated from the quencher end and thus the light emits. The hairpin shape includes two sections, a stem section, and a ring section. The circle or loop section is a single strand which represents the complement of its inputs $X$ and $Y$ while the stem is double strand consists of two sequences $s$ and $s^\prime$, the nine molecular beacons which represent all cases for transformation in the three-step algorithm shown in Table 9.

Also, we need to design the MBs for Not gate as shown for the following Table 10.

| No.of Cycle | Step One       | Step Two       | Step Three   |
|-------------|----------------|----------------|--------------|
| Cycle 1     | $X_1+Y_1$      |                |              |
| Cycle 2     | $X_2+Y_2$      | $T_1+W_1$      |              |
| Cycle 3     | $X_3+Y_3$      | $T_2+W_2$      | $T_1+W_1$    |
| Cycle 4     | $X_4+Y_4$      | $T_3+W_3$      | $T_2+W_2$    |
| Cycle 5     |                | $T_3+W_3$      | $T_2+W_2$    |

### 4. Methods

#### 4.1. Optical simulation of three steps BMSD adder

By comparing this method with the previous one, one could observe that it consumes the and reduces the number of DNA-strands from 93 into 38 (number of nucleotides equal from 558nt to 190 nt). Thus, possible to increase the number of bits in the word and maybe processing a large number of process, we take the same example above for the following addition operation:

\[ X = (1-1101-1)_{BMSD} = (57)_{10}, \quad Y = (10-11100)_{BMSD} = (60)_{10}. \]

The optical simulation of the addition operation as the following Fig. 5.

#### 4.2. Implementation methods

Two approaches have been used to simulate the optical addition operation. These approaches lead immediately to 2D fluorescent processor design that can apply the greet available space-bandwidth producing of an optics system. The following are the two methods for the third design.

---

Fig. 7. The optical simulation of the addition operation in PM. (a) one cycle, (b) two cycle, (c) three cycle, (d) four cycle, and (e) five cycle.
4.2.1. Conventional method (CM)

A simple parallel optical computing for BMSD number addition system is presented. It is a high-speed addition operation because all the values of inputs are processed at the same time, but it takes more numbers of DNA strands, in other word increase the cost of the system. In three steps algorithm, the number of cycles in CM is equal to three cycles without taking into consideration the number of bits in each number. The simulation shows that speed circuit can be achieved by using CM, but a large number of DNA strands is needed. The number of DNA-strands that used is calculated using the following Eq. (5):

\[
St=(n+2)^*M*V
\]  

Eq. (5) can be rewritten in the following form if three operations and each operand contains 15 bits: \( s=(15 + 2)^*3*38=1938 \). For the following addition operation example in CM, here number of cycles \( C \) equal to operation number \( M \):

\[
\begin{bmatrix}
X_1 \\
X_2 \\
X_3 \\
Y_1 \\
Y_2 \\
Y_3
\end{bmatrix} + \begin{bmatrix}
S_1 \\
S_2 \\
S_3
\end{bmatrix} = \begin{bmatrix}
1 - 11 - 10111111111 \_{BMSD} \\
11000 - 1 - 1 - 1101111 \_{BMSD} \\
111111111100 - 1 - 1 - 1 \_{BMSD}
\end{bmatrix} + \begin{bmatrix}
1 - 110 - 1011 - 101 - 1 - 1 \_{BMSD} \\
00 - 1 - 11 - 10011100 - 1 - 1 \_{BMSD} \\
0011 - 1000011111 \_{BMSD}
\end{bmatrix}
\]

\[
\rightarrow \begin{bmatrix}
1 - 11 - 110010100100 \_{BMSD} \\
1 - 101 - 10 - 11011010 - 100 \_{BMSD} \\
100101000010 - 1000 \_{BMSD}
\end{bmatrix}
\]

Addition operation is done in three cycles cycle one, cycle two and cycle three as shown in the following Tables 11, 12, and 13, respectively.

In this design, Eq. (5) is used to calculate the number of DNA strands which is amounting to \((1632)\) of the three periods. The optical simulation of the addition operation using CM is shown in Fig. 6.

4.2.2. Pipeline method (PM)

To obtain a more suitable method for optical arithmetic implementation [38, 39], PM has been applied which acts in several cycles depending on \( m \). In this method, the number of cycles is \( m+2 \). PM reduces the complexity and required hardware (number of DNA strands). The number of DNA strands can be calculated using Eq. (6):

\[
St=(n+2)^*M*V
\]

\[
St: \text{Number of DNA strands.}
\]

\[
n: \text{Number of bits.}
\]

\[
V: \text{Number of DNA strands in Design for one bit.}
\]

Therefore, for this design, Eq. (6) can be rewritten in the following form if three operations and each operand contains 15 bits: \( s=(15 + 2)^*3*38=646 \), in the following simulation above example in PM. Comparing Eq. (5) with Eq. (6), the last one reduces the number of DNA strands by the number of operations (the number of elements in each array). In other word, the number of the used DNA strands is independent on the number of operations, but it depends only on the number of bits \( n \).

Example: · for the example in section (5.1). A number of cycles \( C = M+2 = 5 \). Therefore, the addition process includes five cycles as shown in the following Table 14:-

In this design, Eq. (5) is used to calculate the number of DNA strands which is equal to \((544)\) of the five cycles. The optics simulation needs \( 5 \) five cycles used in PM in the same example in section (5.1) as shown in Fig. 7.

4.3. BMSD multiplication algorithm

Multiplication plays necessary functions in different digital systems such as computers, system controllers, signal processors, and so on. Planning quick multipliers has been a main theoretical and practical importance for computer scientists and engineers [38]. Therefore, several researchers did tensions with rule multiplication-accumulation array (M-A) algorithm [39] and traditional multiplication (Booth’s algorithm) to complete high-radix multiplication [40]. The multiplication of two BMSD number includes two basic processes, generate all partial products (PPs) and accumulate them. Consequently, to improve the speed, we must reduce the time spent in generating PPs and speed up the accumulation. Each PP, can be generated in parallel by the M transformation which is shown in Table 10 [41, 42]. The M-transformation which is used to dispose of the bits of the multiplicand according to a bit of multiplier in the initial step of multiplication. The MSD multiplication can be designed as a separate module [43]. The BMSD multiplication can be composed as a separation module [41]. Multiplication rule shown in Table 15.

![Fig. 8. An example of multiplication of two 4-bits MSD numbers.](image-url)
Fig. 9. The optical simulation of the multiplication operation. (a) partial product $P_0$, $P_1$, $P_2$ and $P_3$. (b) step one, step two and step three for $P_0+P_1$ and $P_2+P_3$. (c) step one, step two and step three for $P_0+P_3$. 
For $X = x_nx_{n-1}...x_1x_0$ and $Y = y_{m-1}...y_1y_0$, the product $P$ is getting by the following Eq. (7):

$$P = XY = \sum_{i=0}^{n-1} PP_i = \sum_{i=0}^{\min(n,m)-1} XY_i$$

(7)

Example:-

For $X = (14)_{10} = (1110)_{BMSD}$ and $Y = (9)_{10} = (1011)_{BMSD}$ then the partial products accumulation is performed as shown in Fig. 8 where $\phi$ denotes a padded zero. At the left, there are four $p_0$, each 8-bits. Then $ps_1$ and $ps_3$ are obtained by adding $P0$ and $P1$, $P2$ and $P3$, respectively. Finally, their product $p$, $ps0$, is got in the same way. We can see that the computations of $ps_j$, $ps_j0$, and $ps02$ are parallel, respectively. Therefore, the MSD multiplication presented is carried out in parallel, and each $pp$ and the product of two n-bit MSD numbers are $(2n)$ and $(2n+1)$ bits, respectively as shown in Fig. 8 [28,33,37].

4.3.1 Enhanced multiplier

Also, the design process goes through the three stages as in addition operation as the following:

4.3.1.1 Stage one: sequence assignment to variable. The same DNA-strands are assigned to the sign digit numbers as in the enhanced adder as shown in Table 3, each DNA-strand composes of 5 nt instead of 6nt [26, 27].

4.3.1.2 Stage two: gate strand design. Take only entries that give active output for truth table which concerned with multiplication operation, each input value represented by DNA-strand code until all the entries that have output are coded, while the two inputs give no active output are ignored. All the select strands are sutured into a single strand, building gate based on multiplication algorithm rule shown in Table 10. The construction of implementation algorithm shown in Table 16.

By converting each variable to its own DNA code, the next sequence is represented the M-transformation.

$3'-TTTTT.CCCCC:TTTTT.GGGGG:AAAAA.CCCCC:AAAAA.GGGGG-5'$

Therefore, the number of DNA strands used are 8DNA-strands (so the number of DNA strands nucleotides that used are 40 nt).

4.3.1.3 Stage three: input strands design. As MBs are designed in Table 4. Therefore, this considers an advantage for presented design in all cases need the same input strands or MBs.

4.3.2 Simulation results in multiplier using enhanced optical processor

To complete multiplication operation, we need to execute the five transformations ($M$, $T$, $W$, $T'$, and $W'$), addition operation in CM and the optical simulation for the same example in the previous design as shown in Fig. 9.

5. Conclusion

DNA strand for constructing arithmetic circuit is considered a new technology emerging in the last decades. This study focuses on using simple principle of design, the simplicity of which is represented by converting truth table rules into the design of adding or multiplication circuit which act in parallelism mechanism. It is important here to refer to the reduction of the used DNA-strands from 93 to 36 and this, in turn, helps to occupy the numbers with greater lengths and thus increase the efficiency of the system. Therefore, it is a fast and efficient arithmetic circuit unit which capable of executing several operations simultaneously in the minimal number of DNA strands. Four computing designs have been proposed for parallel adder and multiplier using BMSD number system in two methods which are CM & PM of implementation. It is clear that the nonexistence of carrying provisions indicates that the idea of greatest and least significant digits is avoided. A test tube of DNA can include trillions of DNA-strands. Every process on a test tube of DNA is achieved on all strands in the tube in parallel about 3* 1014 molecule at a time. Therefore, parallel constructions maybe planned to treat all addition operations all partial products PP $n$ sums. In the future of the DNA optics computer, electric circuits and their wires will be exchanged by biological components, synthesis DNA nitrogen bases in different sequence, depending on required design. Fluorophore particles which are means chromophore molecules that may be found in different color and quencher particles, solutions for the preservation of DNA, and test tubes, will be made the constructed systems to increase effective, more cost productive, higher compact and durability. Finally, we expect a wide spread of studies that support this subject and possibility of storage of the high availability of DNA to use in the storage of information in huge quantities and this is what the world today to cover the need for computers with high capacity to absorb information.

Declarations

Author contribution statement

Qabeela Q. Thabit, Alaa A. Al-Saffar: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

Acknowledgements

I would like to thank Dr. Issa Ahmed Abed for his expert advice and encouragement through this stage and to give me great information about the publication in this journal.

References

[1] A. Srivastava, V. Pandey, Inconsistency in DNA computing and it’s used in cryptography, in: Proceedings 3rd IEEE International Advance Computing Conference (IACC), 2013, pp. 769–774. Gaziabad, India, 22-23 Feb. 2013.
[2] A. Sarker, T. Ahmed, S.M. Mahbubur Rashid, S. Anwar, L. Jaman, N. Tara, Md.M. Alam, H. Md.H. Babu, Realization of reversible logic in DNA computing, in: 11th IEEE International Conference on Bioinformatics and Bioengineering, 2011.
[3] Z. Ezziane, DNA computing: applications and challenges, Nanotechnology 17 (2006) 27–39.
[4] X. Zheng, J. Xu, W. Li, Parallel DNA arithmetic operation based on n-moduli set, Appl. Math. Comput. 212 (2009) 177–184.
[5] S.A. El-Seoud, R. Mohamed, S. Ghoneimy, DNA computing: challenges and application, Int. J. Interact. Mobile Technol. 11 (2) (2017) 74–87.
[6] S. Rinken, V. Ke, Y. Liu, R. Chhabra, H. Yan, Self-assembled DNA nanostructures for distance-dependent multivalent ligand-protein binding, Nat. Nanotechnol. 3 (9) (2008) 418–422.
[7] J. Elbaz, O. Lioubashevski, F. Wang, F. Remacle, R.D. Levine, I. Willner, DNA computing circuits using libraries of DNAzyme subunits, Nat. Nanotechnol. 5 (2010) 417–423.
[8] N.C. Seeman, Nucleic acid junctions and lattices, J. Theor. Biol. 99 (1982) 237–247.
[9] J. Jung, D. Hyun, Y. Shin, Physical synthesis of DNA circuits with spatially localized gates, in: 33rd IEEE International Conference on Computer Design (ICCD), New York, USA, Vol. 5, 2015, pp. 259–265.
[10] R.L. Petersen, M.R. Lakin, A. Phillips, A strand graph semantics for DNA-based computation, Theor. Comput. Sci. 632 (2016) 43–73.
[11] M. Jain, Computer-aided design for DNA self-assembly: process and applications, Int. J. Recent Trends Eng. 1 (2) (2009) 299–303.
