



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 "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 nano-structures 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.

* Corresponding author.

E-mail address: qabelh2010@gmail.com (Q.Q. Thabit).

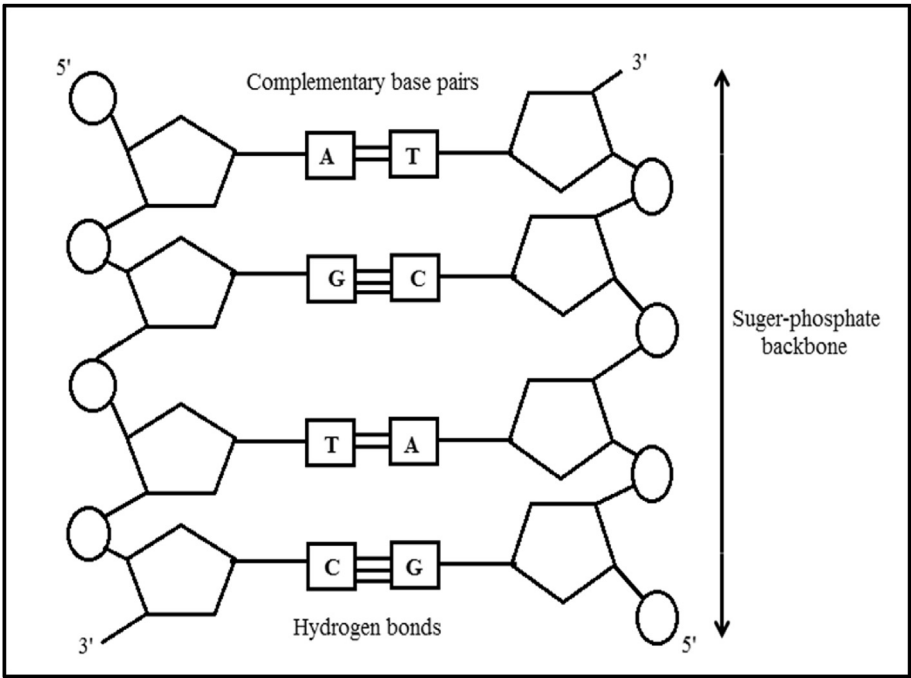


Fig. 1. Double-stranded DNA detailed structure.

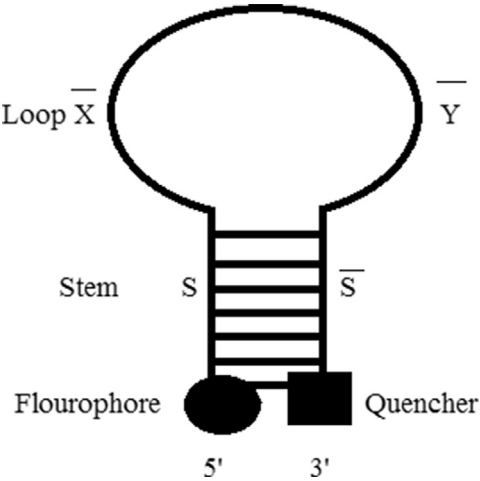


Fig. 2. Input-molecular beacon.

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).

Table 1
Step one T and W Transformation.

X	Y	T_{i+1}	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

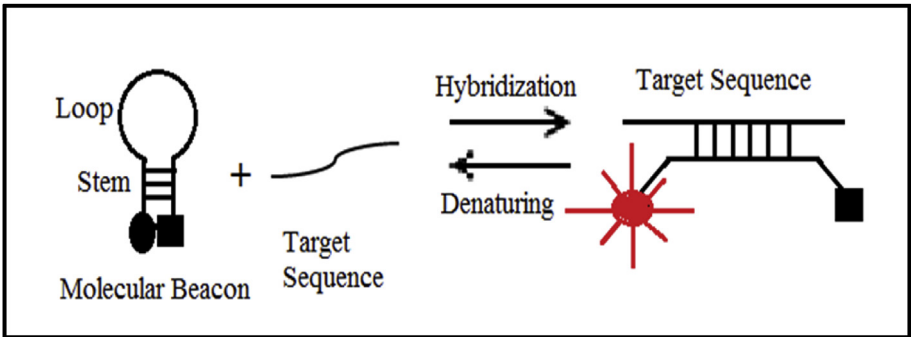


Fig. 3. MB hybridization.

Table 2
Step two T' and W' Transformation.

X	Y	T'_{i+1}	W'_i
-1	-1	-1	0
-1	0	0	-1
-1	1	0	0
0	-1	0	-1
0	0	0	0
0	1	0	1
1	-1	0	0
1	0	0	1
1	1	1	0

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 really used to sense the hybridization success by connecting fluoro-particle with a quencher particle in its two terminals of DNA-hairpin which single-stranded with sequel nucleotides in its 3' and 5' terminals called root section, therefore due to self-binding. they get a construction as given in Fig. 2.

When the MB is in the locked case, it stops dark. This means the fluorophore particle transmits energy and the quencher particle receives it, (darkness as the energy transmitted by the fluorophore particle is received by quencher particle). But, if MB recognizes its target sequence, the ring part hybridizes and result in its loop open up. During this,

fluorophore particle and a quencher particle transfer away creating a change from dark to bright [25] as shown in Fig. 3.

Molecular beacons (MBs) have given their best potential in a category of fundamental researches and effective applications. The first application of molecular beacon MB which uses DNA as it is the cases in this research is the construction of a DNA Computing pattern for Boolean Circuit [26]. The opposite method is the adoption of complementary DNA gates [27], half-adder and half-subtractor design by Molecular beacon principle by C. N. Yang [28], which relies on the influenced hairpin formation property of naphthyridine dimer in a G–G mismatched DNA oligo strand, combined with a generalized gate scheme algorithm [29]. Other fields of research which used DNA, for example, were based on monitoring minimal residual hematologic disease [30], methodologies for genotyping and phenotyping, evaluated in a real-time polymerase chain reaction (PCR) assay to detect the presence of Salmonella species [31], and used in biomedical detection and clinical diagnosis [32].

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]:

$$D = \sum_{j=0}^{n-1} X_j r^j \quad (1)$$

where:-

D :- The decimal number.

X_i :- The i -th digit of BMSD number, $X_i \in \{-\alpha, \dots, -1, 0, 1, \dots, \alpha\}$, $\alpha \leq r-1$.

$(27)_{10}$	1	1	1	0	-1	X	Initial
$(14)_{10}$	1	-1	1	1	0	Y	number
ϕ	0	0	0	-1	1	W_i	Step 1
1	0	1	1	-1	ϕ	T'_{i+1}	
ϕ	1	0	1	1	0	1	W'_i Step 2
0	0	0	0	-1	0	ϕ	T'_{i+1}
$(41)_{10}$	0	1	0	1	0	1	S_i Step 3

Fig. 4. An example of addition of two 5-bits MSD numbers.

Table 3
Sequence assignment of variable.

Variable	Variable Sequence 3'→5'	Complement of Variable Sequence 5'→3'
X^{-1}	TTTTT	AAAAA
X^1	AAAAA	TTTTT
Y^{-1}	CCCCC	GGGGG
Y^1	GGGGG	CCCCC
X^0 or Y^0	ACGTC	TGCAG

Table 4
T-transformation.

T-transformation Sequences		
3'	$X^{-1}.Y^{-1}.X^{-1}.(Y^0 \text{ or } X^0).Y^{-1}.X^1.(Y^0 \text{ or } X^0).Y^1.X^1.Y^1$	5'

Table 5
W-transformation.

W-transformation Sequences		
3'	$X^{-1}.(Y^0 \text{ or } X^0).Y^{-1}.X^1.(Y^0 \text{ or } X^0).Y^1$	5'

Table 6
NOT-gate.

NOT-Transformation Sequence		
3'	$X^1.X^{-1}$	5'

Table 7
T'-transformation.

T'-transformation		
3'	$X^{-1}.Y^{-1}.X^1.Y^1$	5'

Table 8
W'- transformation.

Step Two: W'-transformation		
3'	$X^{-1}.(Y^0 \text{ or } X^0).Y^{-1}.X^1.(Y^0 \text{ or } X^0).Y^1$	5'

Table 9
Input strand sequences.

Inputs		Inputs Symbol		Input derived
X	Y	X	Y	
-1	-1	$\overline{X^{-1}}$	$\overline{Y^{-1}}$	5' CALFluoro S- AAAAA.GGGGG -S BHQ-13'
-1	0	$\overline{X^{-1}}$	$\overline{Y^0}$	5' CALFluoro S- AAAAA.TGCAG -S BHQ-13'
-1	1	$\overline{X^{-1}}$	$\overline{Y^1}$	5' S-AAAAA.CCCCC-S BHQ-13'
0	-1	$\overline{X^0}$	$\overline{Y^{-1}}$	5' CALFluoro S- TGCAG.GGGGG -S BHQ-13'
0	0	$\overline{X^0}$	$\overline{Y^0}$	5' XXXS- TGCAG. TGCAG -S BHQ-13'
0	1	$\overline{X^0}$	$\overline{Y^1}$	5' FAM S- TGCAG.CCCCC -S BHQ-13'
1	-1	$\overline{X^1}$	$\overline{Y^{-1}}$	5' S-TTTTT. GGGGG-S BHQ-13'
1	0	$\overline{X^1}$	$\overline{Y^0}$	5' FAM S- TTTTT. TGCAG -S BHQ-13'
1	1	$\overline{X^1}$	$\overline{Y^1}$	5' FAM S- TTTTT.CCCCC -S BHQ-13'

Table 10
NOT-gate input strand sequence.

Inputs	Inputs Symbol	Input derived (MB)
X	X	
-1	$\overline{X^{-1}}$	5' FAM S-AAAAA -S BHQ-13'
0	$\overline{X^0}$	5' S-TGCAG -S BHQ-13'
1	$\overline{X^1}$	5' CALFluoro S-TTTTT -S BHQ-13'

r :- The radix of the BMSD number system.

n :- Number of a (bits in each operand) digit in BMSD number.

Suppose the binary modified signed-digit BMSD representation of augends X and added Y are [35]:

$$X_{\text{BMSD}} = X_{n-1}, \dots, X_i, \dots, X_0 \text{ and}$$

$$Y_{\text{BMSD}} = Y_{n-1}, \dots, Y_i, \dots, Y_0.$$

The three steps addition is performed according to the following [35, 36]:-

Step one: compute sum (W) and carry (T) according to the following Eq. (2)

$$X_i + Y_i = 2T_{i+1} + W_i \quad (i = 0, \dots, n-1) \quad (2)$$

Where T: is the carry and

W: is the sum.

n: Number of bits in each operand. The rule of step one shown in Table 1.

Step Two: This step compute sum (W') and carries (T') according to the following Eq. (3). The rule of step two shown in Table 2.

$$T_i + W_i = 2T'_{i+1} + W'_i \quad (i = 0, \dots, n-1) \quad (3)$$

Step Three: computes sum according to the following Eq. (4)

$$S_i = W'_i + T'_i \quad (i = 0, \dots, n-1) \quad (4)$$

Example:-

The addition operation in a three-step algorithm for two BMSD numbers $X = (1110-1)_{\text{BMSD}} = (27)_{10}$ and $Y = (1-1110)_{\text{BMSD}} = (14)_{10}$ as shown in Fig. 4.

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 depends 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} and put in array which begins from 3' end and finishes in 5' end as the following: the strands, connected to the variable, are directly applied, 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 $T_{i+1} = -1$ is deemed and its X value (-1) and Y value (0) and added to array which becomes (-1,-1,-1,0). Eventually, the following row4 with $T_{i+1} = -1$ is considered and its X value (0) is compared with the last value put in the array (0). Therefore, we take Y

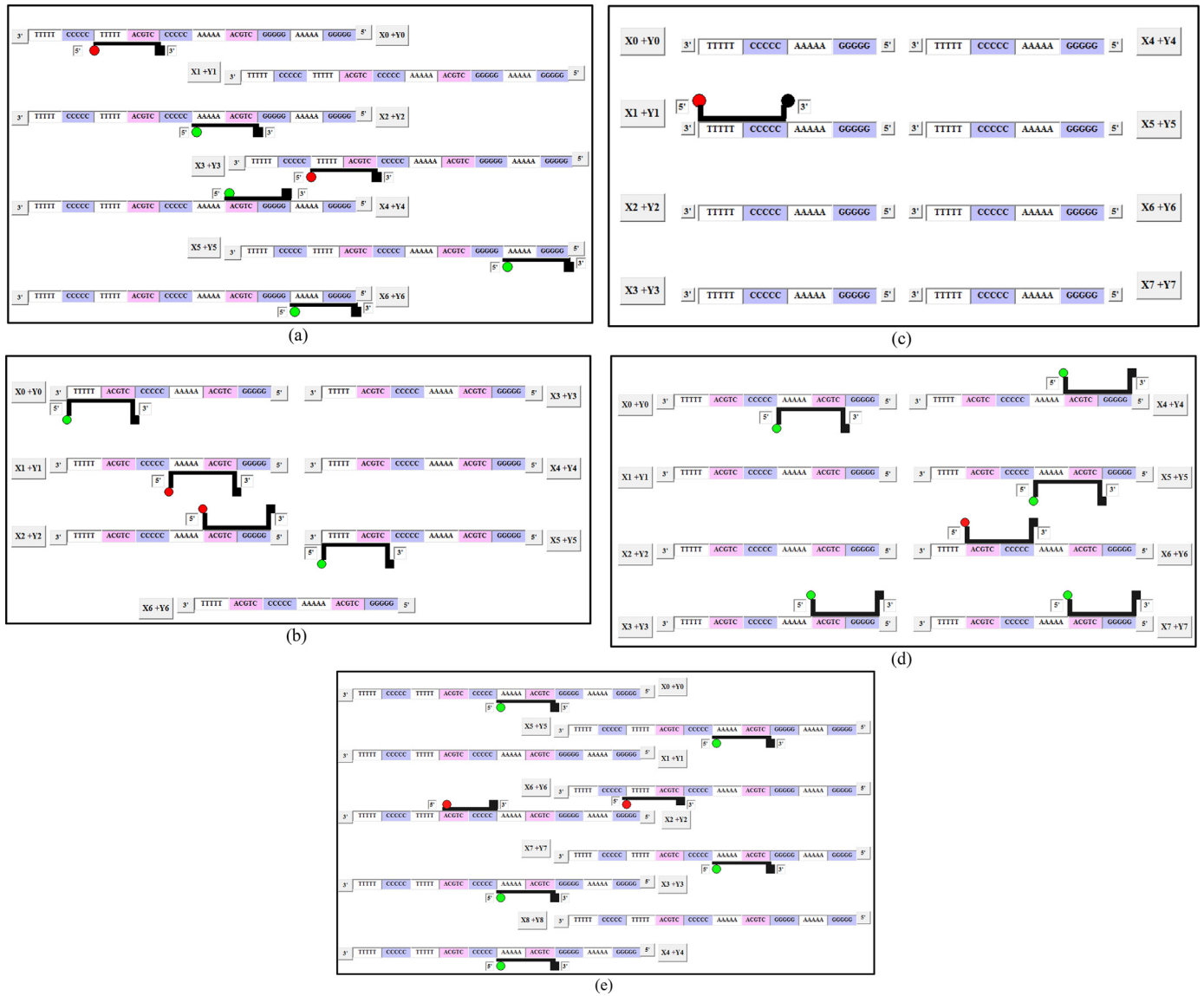


Fig. 5. The optical simulation of addition operation. (a) Step-one (T-transformation). (b) Step-one (W-transformation). (c) Step-two (T'-transformation). (d) Step-two (W'-transformation). (e) Step-three (T-transformation).

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,-1,1,1,0,1). Therefore, we need 10 DNA-strand (50nt) for achieving

Table 11

Cycle one results.

Operation	T	W
(X ₁ +Y ₁)	1-11-1-111101101100	000011-1000-100-100
(X ₂ +Y ₂)	11-1-11-1-1-111111000	0-1-111-10110-10-1-100
(X ₃ +Y ₃)	1111011111110000	0-1-1000-1-1-1-10-1-1000

Table 12

Cycle two results.

Operation	T'	W'
(T ₁ +W ₁)	0000100000000000	1-11-1000101001000
(T ₂ +W ₂)	0-101-100100000000	100000-1001010-100
(T ₃ +W ₃)	000000000000000000	100101000010-1000

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'TTTT.CCCC:TTTTC.ACGTC.CCCCC: AAAAA.ACGTC.GGGG-G:AAAAA.GGGGG-5'

3.2.2. W-transformation

Scanning all rows, the first one results in active output W_i is row 2, we take the two inputs (-1, 0) and save in array, then the following row4 with W_i = 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 13

Cycle three results.

Operation	T (Final Sum)
(T' ₁ + W' ₁)	1-11-1100101001000
(T' ₂ + W' ₂)	1-101-10-1101010-100
(T' ₃ + W' ₃)	100101000010-1000

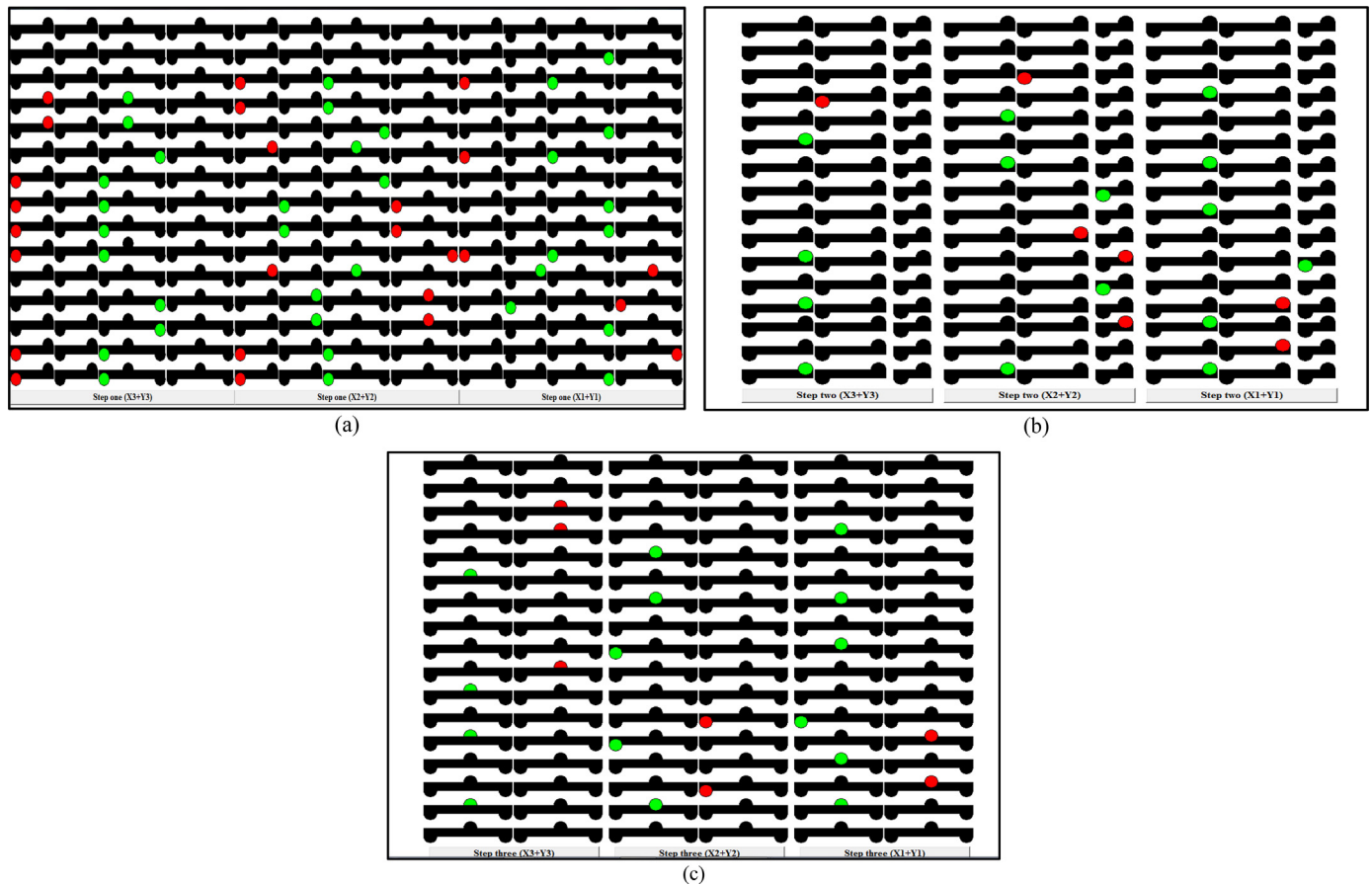


Fig. 6. The optical simulation of the addition operation in CM. (a) one cycle. (b) two cycles. (c) three cycles.

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'- TTTT. 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: TTTT-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 $\{(-1 + -1)\}$. 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'-a 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'- TTTT.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'-a 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'- TTTT. ACGTC.CCCCC: AAAAA. ACGTC.GGGGG -5'

3.2.6. Step three (T-transformation)

Step three include 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

Table 14

Five cycles for three-step of the addition operation in the PM.

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	————	T_3+W_3	$T'_2 + W'_2$
Cycle 5	————	————	$T'_3 + W'_3$

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 $\bar{X}$, and $\bar{Y}$ while the stem is double strand consists of two sequence s and $\bar{s}$, 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.

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 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:-

$X = (1-11101-1)_{\text{BMSD}} = (57)_{10}$, and $Y = (10-11100)_{\text{BMSD}} = (60)_{10}$, 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.

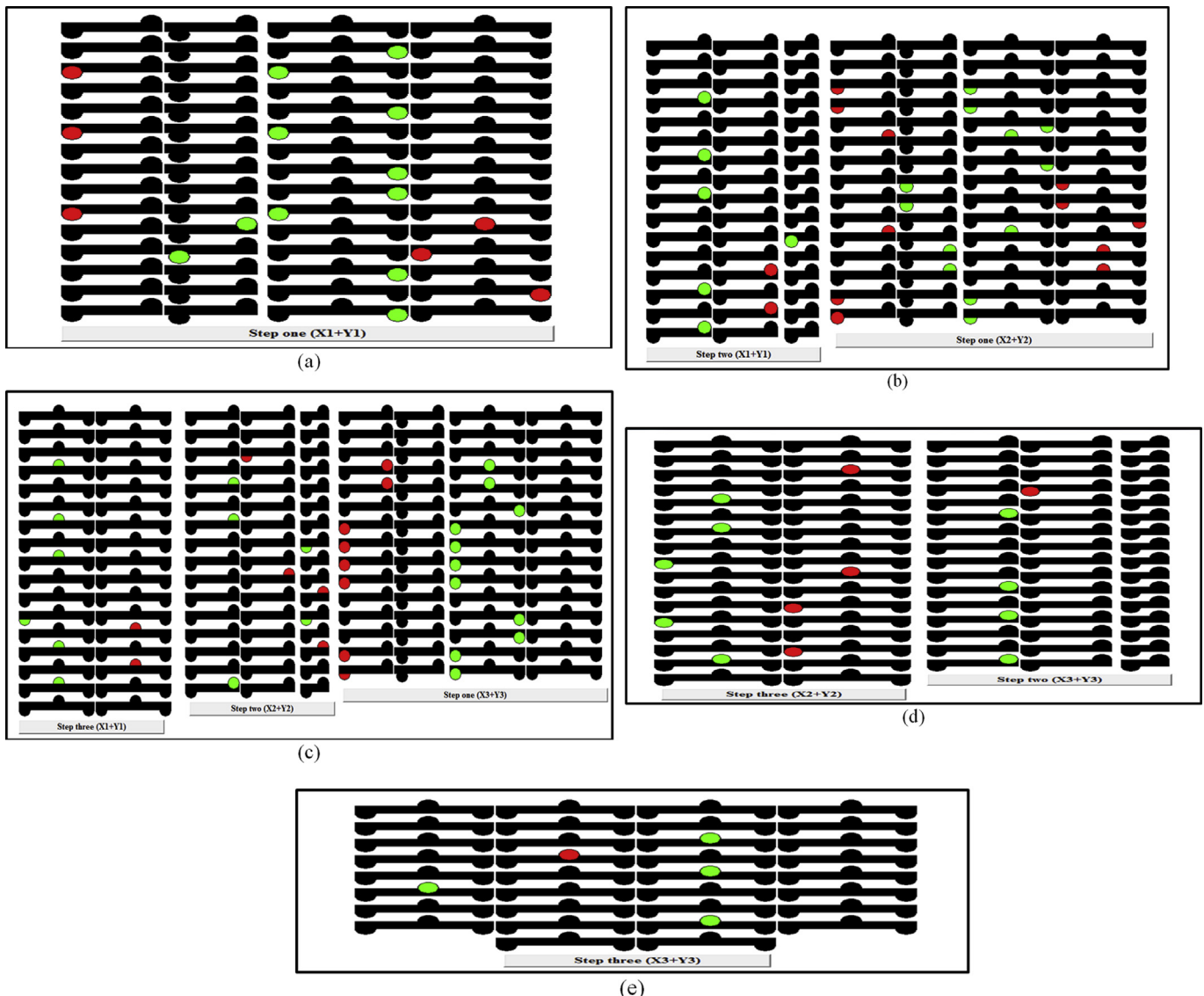


Fig. 7. The optical simulation of the addition operation in PM. (a)one cycle, (b) two cycle, (c) three cycle, (d) four cycle, and (e) five cycle.

Table 15

Truth table of M-transformation.

M	-1	0	1
-1	1	0	-1
0	0	0	0
1	-1	0	1

4.2.1. Conventional method (CM)

A simple parallel optical computing for BMSD number addition system is presented. It is 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):-

$$St=(n+2)*M*V \quad (5)$$

St: Number of DNA strands.

n: Number of bits in each operand.

M: Number of operations.

V: Number of DNA strands in Design

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:-

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} 1-11-1011111111_{BMSD} \\ 11000-1-1-1101111_{BMSD} \\ 111111111100-1-1-1_{BMSD} \end{bmatrix} + \begin{bmatrix} 1-110-1011-101-101-1_{BMSD} \\ 00-1-1-11-10011100-1-1_{BMSD} \\ 0011-1000011111_{BMSD} \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 1-11-1100101001000_{BMSD} \\ 1-101-10-1101010-100_{BMSD} \\ 100101000010-1000_{BMSD} \end{bmatrix}$$

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):-

$$St=(n+2)*V \quad (6)$$

St: Number of DNA strands.

n: Number of bits.

V: 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:-

$St = (15 + 2)*38 = 646$, 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 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_i 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.

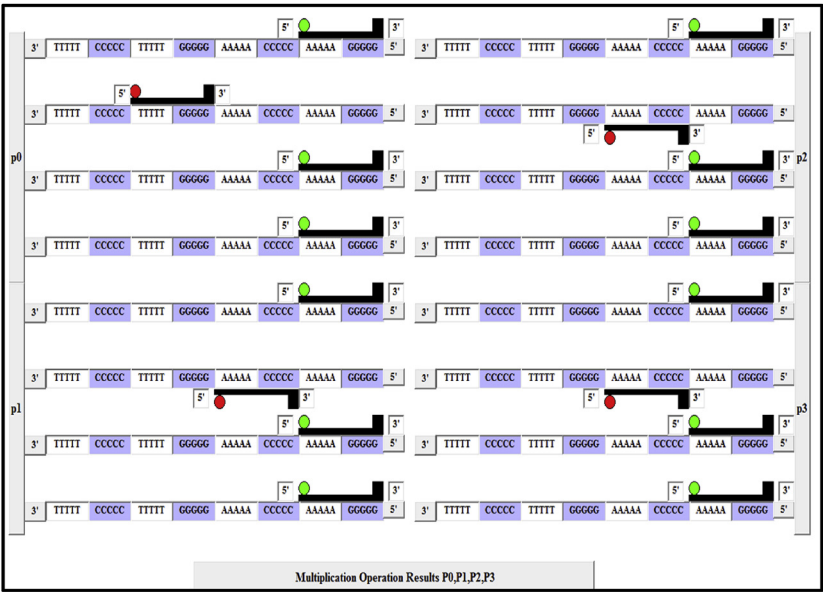
Table 16

M-Transformation.

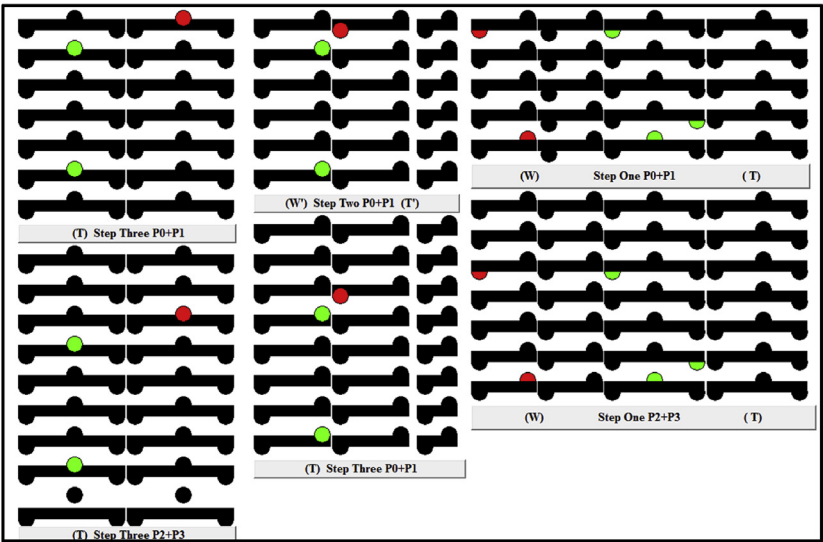
	M-transformation Sequence	
3'	$X^{-1}.Y^{-1}; X^{-1}.Y^1; X^1.Y^{-1}; X^1.Y^1$	5'

$$\begin{array}{l} P_0: \phi\phi\phi\phi\bar{1}\bar{1}\bar{1}0 \\ P_1: \phi\phi\phi\phi1110\phi \\ P_2: \phi\phi0000\phi\phi \\ P_3: \phi1110\phi\phi\phi \end{array} \left. \begin{array}{l} + \phi\phi1\bar{1}0\bar{1}10 \\ + \phi\phi100\bar{1}0000 \end{array} \right\} + (1\bar{1}00000\bar{1}0)_{MSD} = (126)_{10}$$

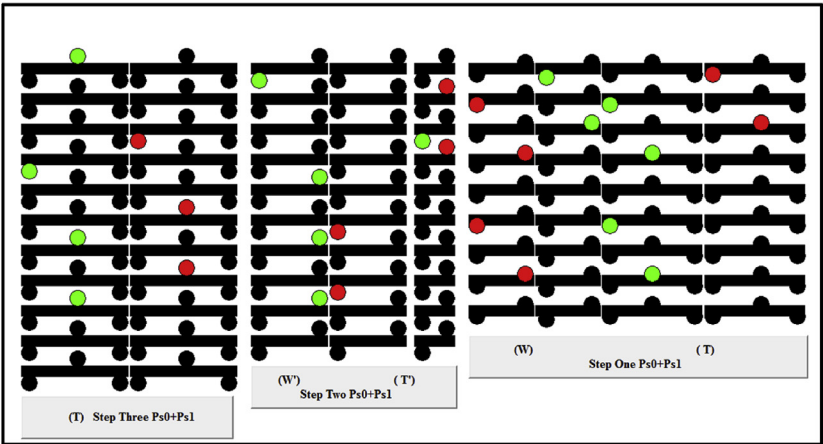
Fig. 8. An example of multiplication of two 4-bits MSD numbers.



(a)



(b)



(c)

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_1 .

For $X = X_{n-1} \dots X_1 X_0$, and $Y = Y_{n-1} \dots Y_1 Y_0$, the product P is getting by the following Eq. (7):-

$$P = XY = \sum_{i=0}^{n-1} PP_i = \sum_{i=0}^{n-1} XY_i \quad (7)$$

Example:-

For $X = (14)_{10} = (1110)_{\text{BMSD}}$, and $Y = (9)_{10} = (1011)_{\text{BMSD}}$ then the partial products accumulation is performed as shown in Fig. 8 where ϕ denotes a padded zero. At the left, there are four p_i s, each 8-bits. Then ps_1^0 and ps_1^1 are obtained by adding P_0 and P_1 , P_2 and P_3 , respectively. Finally, their product p , ps_2^0 , is got in the same way. We can see that the computations of p_i , psj_1 , and ps_2 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, and 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 selection 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 3×10^{14} molecule at a time. Therefore, parallel constructions maybe planned to treat all addition operations all partial products PP's 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 its used in cryptography, in: Proceedings 3rd IEEE International Advance Computing Conference (IACC), 2013, pp. 769–774. Ghaziabad, 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. Eziane, 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. Rinker, Y. 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.

- [12] C. Pistol, V. Mao, V. Thus, A.R. Lebeck, C. Dwyer, Encoded multichromophore response for simultaneous label-free detection, *Small* 6 (7) (2010) 843–850.
- [13] J. Panga, A.R. Lebeck, C.L. Dwyer, Modeling and simulation of a nanoscale optical computing system, *J. Parallel Distrib. Comput.* 74 (2014) 2470–2483.
- [14] C. LaBoda, H. Duschl, C.L. Dwyer, DNA-enabled integrated molecular systems for computation and sensing, *Acc. Chem. Res.* 47 (2014) 1816–1824.
- [15] R. Laddha, K. Honwadkar, Digital data storage on DNA, *Int. J. Comput. Appl.* 142 (2) (2016) 43–46.
- [16] A. Swati, F. Mathuria, S. Bhavani, E. Malathy, R. Mahadevan, A review on various encoding schemes used in digital DNA data storage, *Int. J. Civ. Eng. Technol.* 8 (12) (2017) 108–114. https://www.iaeme.com/MasterAdmin/uploadfolder/IJCIET_08_12_013/IJCIET_08_12_013.pdf.
- [17] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E.M. Leproust, B. Sipos, E. Birney, Towards practical, high-capacity, low-maintenance information storage in synthesized DNA, *Nature* 494 (2013) 77–80.
- [18] L. Sennels, T. Bontin, To DNA, all Information is Equal, *Artif. DNA PNA XNA* 3 (3) (2012) 109–111.
- [19] G.M. Church, Y. Gao, S. Kosuri, Next-generation digital information storage in DNA, *Science* 337 (2012) 1628–1629.
- [20] Alaa A. Al-Saffar, Qabeela Q. Thabit, Optical computing for A three-step binary modified signed-digit addition using DNA, in: *Fourth International Conference on Computational Science and Technology*, Kuala Lumpur, Malaysia, 29–30, Nov, 2017.
- [21] Alaa A. Al-Saffar, Qabeela Q. Thabit, Simulation of nanoscale optical signed digit addition based on DNA-strands, in: *Proceedings International Conference on Advanced Science and Engineering (ICOASE)*, Kurdistan Region, Iraq, 2018, pp. 128–133.
- [22] M. Amos, G. Paun, G. Rozenberg, A. Salomaa, Topics in the theory of DNA computing, *Theor. Comput. Sci.* 287 (2002) 3–38.
- [23] A. Sarker, T. Ahmed, S.M.M. 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*, Taichung, Taiwan, 2011, pp. 261–265, 24–26 Oct.
- [24] A. Klara, H. Jeffrey, M. Ron, M. Barbara, M. Carolyn, S. George, R. John, Molecular beacons as diagnostic tools: technology and applications, *Clin. Chem. Lab. Med.* 4 (4) (2003) 468–474.
- [25] K. Boruah, J. Ch. Dutta, DNA computing algorithm for realization of DNA boolean logic based on micro-cantilever deflection, *Int. J. Pharm. Sci. Rev. Res.* 42 (1) (2017) 111–116. <http://globalresearchonline.net/journalcontents/v42-1/21.pdf>.
- [26] K. Boruah, J.C. Dutta, Development of a DNA computing model for boolean circuit, in: *International Conference on Advances in Electrical, Electronics, Information, Communication, and Bio-Informatics (AEEICB16)*, Chennai, TamilNadu, India, 2016.
- [27] B.S.E. Zoraida, M. Arocka, B.S.M. Ronald, R. Ponalagusamy, A novel generalized design methodology and realization of boolean operations using DNA, *Biosystems* 97 (2009) 146–153.
- [28] C. Yang, C. Hsu, Y. Chuang, Molecular beacon based half adder and half subtractor, *Chem. Commun.* 48 (2012) 112–114.
- [29] K. Boruah, J.C. Dutta, An improved generalized DNA computing model to simulate logic functions and combinational circuits, *Int. Inf. Technol.* 10 (3) (2018) 379–390. Springer, <https://link.springer.com/article/10.1007/s41870-018-0110-7>.
- [30] C. Wei, J.H. Lipton, S.K. Reid, Monitoring of Minimal Residual Hematologic, *Cell and Tissue Based Molecular Pathology*, 2009, pp. 135–144.
- [31] S.H. Liming, A.A. Bhagwat, Application of molecular beacon-real-time PCR technology to detect *Salmonella* species contaminating fruits and vegetables, *Int. J. Food Microbiol.* 95 (2004) 177–187.
- [32] Y. Kim, D. Sohn, W. Tan, Molecular beacons in biomedical detection and clinical diagnosis, *Int. J. Clin. Exp. Pathol.* 1 (2008) 105–116. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2480550/>.
- [33] R.S. Fyath, A.A.W. Alsaffar, M.S. Alam, Nonrecorded trinary signed-digit multiplication based on digit grouping and pixel assignment, *Opt. Commun.* 230 (2004) 35–44.
- [34] R.S. Fyath, A.A.W. Alsaffar, M.S. Alam, Optical binary logic gate-based modified signed-digit arithmetic, *Opt. Laser. Technol.* 34 (2002) 501–508.
- [35] K. Boruah, J. Ch. Dutta, DNA computing model for realization of boolean circuit, *Int. J. Control Theory Appl.* 9 (21) (2016) 281–287. https://www.researchgate.net/publication/316599451_DNA_computing_model_for_realization_of_boolean_circuit.
- [36] H. He, J. Peng, Y. Liu, X. Wang, K. Song, Research and Design of A MSD Adder of Ternary Optical Computer, Springer- V ICAIC-228, 2011, pp. 413–420.
- [37] G.K. Maity, S.P. Maity, J.N. Roy, MZI based modified trinary number system, *Sci. Direct* 4 (2012) 297–302.
- [38] N. Takagi, H. Yasuura, S. Yajima, High-speed VLSI multiplication algorithm with a redundant binary addition tree, *IEEE Trans. Comput.* c-34 (9) (1985) 789–796.
- [39] A. Persson, L. Bengtsson, Forward and reverse converters and moduli set selection in signed-digit residue number systems, *J. Sign. Process. Syst.* 56 (2009) 1–15. <https://link.springer.com/article/10.1007/s11265-008-0249-8>.
- [40] A. Omondi, Residue Number Systems Theory and Implementation, Imperial College Press, 1, 2007. <https://epdf.tips/residue-number-systems-theory-and-implementation.html>.
- [41] Y. Jin, X. Wang, J. Peng, M. Li, Z. Shen, S. Yang, Vector-matrix multiplication in ternary optical computer, *Int. J. Numer. Anal. Model.* 9 (2) (2012) 401–409. <http://www.math.ualberta.ca/ijnam/Volume-9-2012/No-2-12/2012-02-20.pdf>.
- [42] X. Wang, P.J. Jie, J. Yi, L. Mei, Z. Shen, O. Shan, Vector-matrix multiplication based on ternary optical computer, in: *Proceeding of the 2nd International Conference on High-Performance Computing and Application*, Shanghai University, China, 2009.
- [43] S.S. Alsheraidah, Designing an optical multiplier by using a modified signed-digit number system, *J. Basrah Res. (Sci.)* 35 (3) (15th June 2009) 15–22. <https://www.iasj.net/iasj?func=fulltext&aid=57095>.