American Academic Scientific Research Journal for Engineering, Technology, and Sciences

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

https://asrjetsjournal.org/index.php/American_Scientific_Journal/index

MSDRAM: Multivalued Sequence Storage of Random Access Memory Using DNA Technology

Tamanna Tabassum [*]

Ahsanullah University of Science and Technology, 141 & 142, Love Road, Dhaka-1208, Bangladesh

Email: tamanna.cse@aust.edu

Abstract

The rapid growth in data generation demands innovative solutions for efficient storage and retrieval, far beyond the capabilities of traditional silicon-based Random Access Memory (RAM). DNA-based storage systems have emerged as a revolutionary approach, leveraging DNA's intrinsic properties such as high density, stability, and scalability. Unlike binary encoding, ternary RAM leverages the quaternary nature of DNA bases to represent multivalued data, thereby enhancing storage density and computational efficiency. This technology achieves unprecedented storage densities by mapping multivalued data to synthetic DNA sequences while implementing advanced biochemical techniques for storage. This paper introduces MSDRAM (Multivalued Sequence Storage of Random Access Memory), a novel architecture utilizing DNA technology to overcome the limitations of conventional storage systems. This proposed research sets the foundation for hybrid storage architectures, combining the strengths of molecular and silicon-based technologies to meet future computational demands. The proposed architecture of multivalued SDRAM demonstrates that it achieves a storage density of a single petabyte per gram of DNA, detailing its encoding unit, DNA-based storage medium, and access mechanisms that significantly outperform traditional RAM and binary-based DNA RAM in capacity and heat efficiency. This research highlights the potential of DNA technology for scalable, energy-efficient memory systems and addresses the challenges of heat, speed, and environmental sensitivity.

Keywords: DNA Sequence; Molecule; Random Access Memory; Multi-sequenced; Qutrit; MSDRAM.

Received: 10/31/2024
Accepted: 12/31/2024
Published: 1/10/2025

----

[*] Corresponding author.

1. Introduction

Humans have always been archivists, preserving information throughout history. In the digital age, the accumulation of vast amounts of data has reached unprecedented levels [1, 2]. Cisco, a global technology conglomerate, predicted that by 2019, data generated by the "Internet of Everything" (or the Internet of Things) would exceed 500 zettabytes [3, 4]. This exponential growth has been further driven by the widespread adoption of personal cloud storage. However, data storage relies heavily on silicon, a finite resource that is rarely found in its pure form [5]. Researchers predict that by 2040, the global supply of memory-grade silicon will be exhausted Reference [6, 7]. To address this impending challenge, DNA-based data storage has emerged as a promising alternative [8, 9]. DNA, composed of nucleotides—adenine (A), thymine (T), cytosine (C), and guanine (G)—is nature's information storage medium, capable of storing data through the sequence of these bases [10, 11]. DNA can be synthesized (written), sequenced (read), and duplicated with exceptional accuracy and stability. Its durability is evident from the successful genome sequencing of a fossilized horse that lived over 500,000 years ago [12]. DNA storage also requires minimal energy for long-term maintenance, further enhancing its practicality Reference [13].

The most significant advantage of DNA storage lies in its unparalleled density, far surpassing that of traditional electronic storage systems [14]. For instance, Escherichia coli exhibits a storage density of approximately 1019 bits per cubic centimeter, as calculated in a 2016 Nature Materials study by George [15]. At such a density, the world's annual data storage requirements could be met by a DNA cube measuring just one meter on each side. Despite its enormous potential, challenges remain in making DNA-based storage widely accessible. The cost and speed of synthesizing and sequencing DNA must be significantly reduced to compete with conventional storage systems [16, 17]. Even if DNA does not replace all existing storage technologies, it will undoubtedly play a critical role in large-scale information generation and long-term archival storage [18]. DNA memory storage offers revolutionary capabilities, addressing archival data needs at an Exabyte scale, far beyond the reach of current technologies [19, 20]. Chemically, DNA is an ideal long-term storage solution due to its molecular stability. When dried, it can maintain integrity for tens of thousands of years [21].

Currently, synthetic DNA has achieved an extraordinary storage density of 215 petabytes per gram, marking a breakthrough in data storage technology [21, 22].

DNA computers have the potential to surpass the computational power of conventional digital systems, but several technical challenges must be addressed for their successful implementation [23]. DNA computing operates using biomolecules and biochemical reactions to execute computational algorithms, leveraging biotechnological techniques for their design and evolution [24]. As an emerging paradigm, DNA computing is often regarded as a promising alternative to solid-state computers due to its unique features, though it also introduces fundamental limitations.

Michael Conrad emphasized the complementary qualities of computing both quantum and DNA, highlighting the trade-off between efficiency of computation, evolutionary versatility, and programmability [25, 26]. Programmability refers to the ability of a system to execute instructions accurately and efficiently, while computational efficiency describes how effectively interactions within the system contribute to the overall computation. Evolutionary adaptability signifies the ability of the system to operate in dynamic and unpredictable circumstances. Traditional digital, quantum, and biomolecular systems perform well in these domains while falling short in others. Biomolecular computers, inspired by the adaptability and resilience of biological systems, can evolve using molecular biology techniques and enzymes, allowing them to adapt to environmental changes and inputs [27]. This adaptability has influenced the development of evolutionary programming and artificial neural networks. However, programming biomolecular computers with precision remains a significant challenge due to the inherent complexity of biochemical interactions.

Additionally, these systems suffer from low computational efficiency, as many interactions are error-prone or fail to contribute directly to the intended result [28, 29]. In conclusion, while DNA computers offer promising capabilities, particularly in adaptability and molecular-level computation, overcoming challenges related to programming accuracy and computational efficiency is essential for their practical realization.

Sequential circuits rely on the presence of memory to perform their operations. Flip-flop is a basic memory element capable of storing a single bit of information. At the same time, a register extends this capacity by storing a single “word,” typically consisting of 32 to 64 bits. Random access memory (RAM) is employed to store larger volumes of data. RAM provides temporary, volatile storage, enabling the data transfer to or from any memory location with consistent access time, irrespective of the address location. Each memory cell within RAM is assigned an address that specifies the stored data, which can consist of multi-bit words. A functional RAM system must meet three key requirements: it should be capable of storing multiple words (one per address), retrieving the word stored at a specific address, and modifying the stored word at a given address. In the context of multivalued memory, ternary logic offers an alternative approach. Ternary logic systems operate with three possible states (e.g., 0, 1, and 2) instead of binary logic's two states. In ternary DNA computing, two DNA sequences function as inputs, while a single DNA sequence serves as the output. The detection of these DNA sequences relies on fluorescence levels. When particular molecules immediately absorb visible light spectrum electromagnetic wavelengths, they discharge light with a diminished energy level, a phenomenon known as fluorescence. This process enables accurate identification of DNA sequences, facilitating ternary DNA computations.

2. Literature Study

This section provides a detailed discussion on the fundamentals of multivalued memory, basic gate operations, and ternary-based storage RAM.

2.1. Multivalued DNA Computing

The invention of DNA-based storage involves techniques for storing, processing, and selectively retrieving data encoded within sequence-controlled polymer barcoded nanoparticles. DNA memory storage is implemented using DNA origami, structured DNA objects, and encapsulated DNA memory, which are organized into memory blocks with unique ID tags. These memory blocks enable the rapid retrieval of large-scale data through associative memory and Boolean logic operations, significantly reducing the read-out time required during DNA sequencing while ensuring efficient archival storage of information. Furthermore, optical barcoding of DNA memory packets, whether on DNA structures or encapsulated DNA memory, allows for high-speed sorting of molecularly encoded data at rates exceeding 100 Mb/s [30}. To enhance search efficiency, additional biochemical barcodes facilitate algorithmic sorting, further narrowing the scope of data retrieval.

2.2. Random Access Memory (RAM)

The primary function of RAM is to provide fast read and write access to data storage. RAM performs two key operations: write and read. The write operation specifies data transfer into memory, while the read operation specifies data transfer out of memory. Once one of these control signals is accepted, the internal circuits within the memory execute the requested operation. To store a new word in memory, the following steps are performed:

- The binary address of the target memory location is applied to the address lines.
- The data bits to be stored are applied to the data input lines.
- The write signal is activated.

At this point, the memory unit stores the data bits from the input lines into the specified memory location. To retrieve a stored word from memory, the process involves:

- Applying the binary address of the desired memory location to the address lines.
- Activating the read signal.

The memory unit transfers the bits of the selected word from the specified address to the output data lines. Notably, the contents of the selected memory location remain unaltered after the read operation.

A block diagram of a ternary (base-3) RAM is illustrated in Fig 1. It comprises k input lines and n output lines. Each unique bit combination of the input variables represents an address, whereas a word is the bit combination that is generated on the output lines. The number of output lines, n, is equivalent to the number of bits in a word. An address is a ternary number corresponding to one of the k variables' minterms.

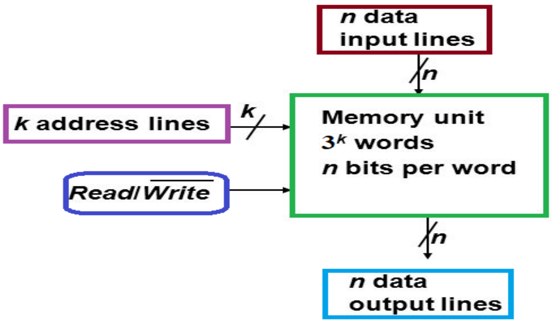

Figure 1: Block Diagram of 3^k-to-n RAM

The n data input lines deliver the information to be stored in the memory, while the n data output lines provide the information retrieved from a specific word among the 3^k available within the memory. This memory is referred to as a 3^k-to-n memory, where k address lines are used to specify one of the 3^k possible addresses. Each address corresponds to an n-bit word. For instance, a $3^2 \times 1$ RAM contains $3^2 = 9$ words, with each word being 1 bit in length. The RAM requires 2 address lines, and the total storage capacity is calculated as $3^2 \times 3^0 = 3^2$ bits.

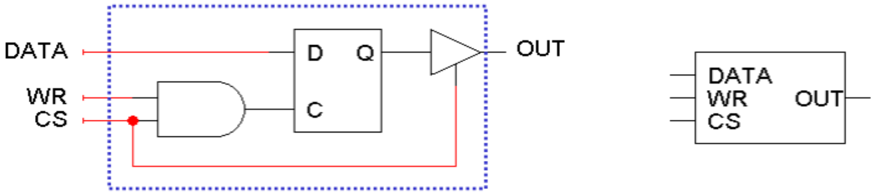

Figure 2: Block Diagram of RAM Chip

The block diagram (Fig 2) outlines the primary interface of a RAM system. A Chip Select (CS) signal enables or disables the RAM, while the DATA input specifies the memory address for reading or writing operations. The WR signal determines whether the operation is a read or write. WR is set to 0 to perform a read operation, and the OUT line will display the n-bit value stored at the specified DATA. For a write operation, WR is set to 1, and the DATA input provides the n-bit value to be stored at the selected address. A single-bit RAM cell can be implemented using a latch to store each bit. In this case, a DATA input is unnecessary since it represents only one bit of memory.

Writing to the RAM cell: When CS = 1 and WR = 1, the latch control input is activated, and the DATA input value is stored in the D latch.

Reading from the RAM cell: When CS = 0 or WR = 0, the latch control input remains inactive, maintaining the current state of the latch. The stored value appears on OUT when CS = 1.

By combining these single-bit cells, a 9-to-1 RAM can be constructed. In this configuration, DATA requires three bits to select one of the nine words. Each word is a single bit, making both DATA and OUT one bit wide. Word selection is achieved using a decoder connected to the CS inputs of the RAM cells, ensuring that only one cell is read from or written to at any given time.

3. Multivalued Random Access Memory (MSDRAM)

Initially, the multivalued quantum RAM is a combinational circuit with multivalued quantum AND operations connected as a multivalued quantum decoder, multivalued molecular RAM cells, and several multivalued quantum OR operations to the outputs in the unit. With k address lines and n output lines in RAM, the output functions will be calculated through the sum of the minterms form. For a RAM with (k = 2) input variables and (n=1) output bit, the circuit is called a 9-to-1 multivalued DNA-based RAM. In the case of a 9-to-1 RAM, the general organization (Fig 3) includes 9 words corresponding to 1 output sequence. Here, sequence (A) and sequence (B) can each assume one of three possible values: ACCTAG (representing 0), CAAGCT (representing

1), or TGGATC (representing 2). This arrangement allows 9 unique word sequences to be stored in the RAM unit. The RAM features a single output line, Z1, where a specific word sequence is selected based on the input values applied to the 2 input lines. Since 32 = 9, the 3 possible input sequences enable the specification of 9 addresses. To implement these minterms, the multivalued DNA 9-to-1 RAM requires a multivalued DNA 2-to-9 decoder along with multivalued DNA OR operations.

3.1 Fundamental Component Structure of MSDRAM

To perform the operations of a ternary DNA-based 9-to-1 RAM, the following components are required: a 2-to-9 DNA decoder, multivalued sequence RAM cells, and DNA OR operations to implement the corresponding minterms.

3.1.1 Multivalued Molecular RAM Cell

- ***Circuit Design of Molecular Cell:***

The molecular cell design is fundamentally based on the R-S flip-flop (Fig 4), with three input sequences labeled "Select," "R/W," and "Input," and a single output labeled "M0." The implementation requires two DNA NOT, six DNA AND, and two DNA NOR operations. The steps for generating the molecular cell output are as follows:

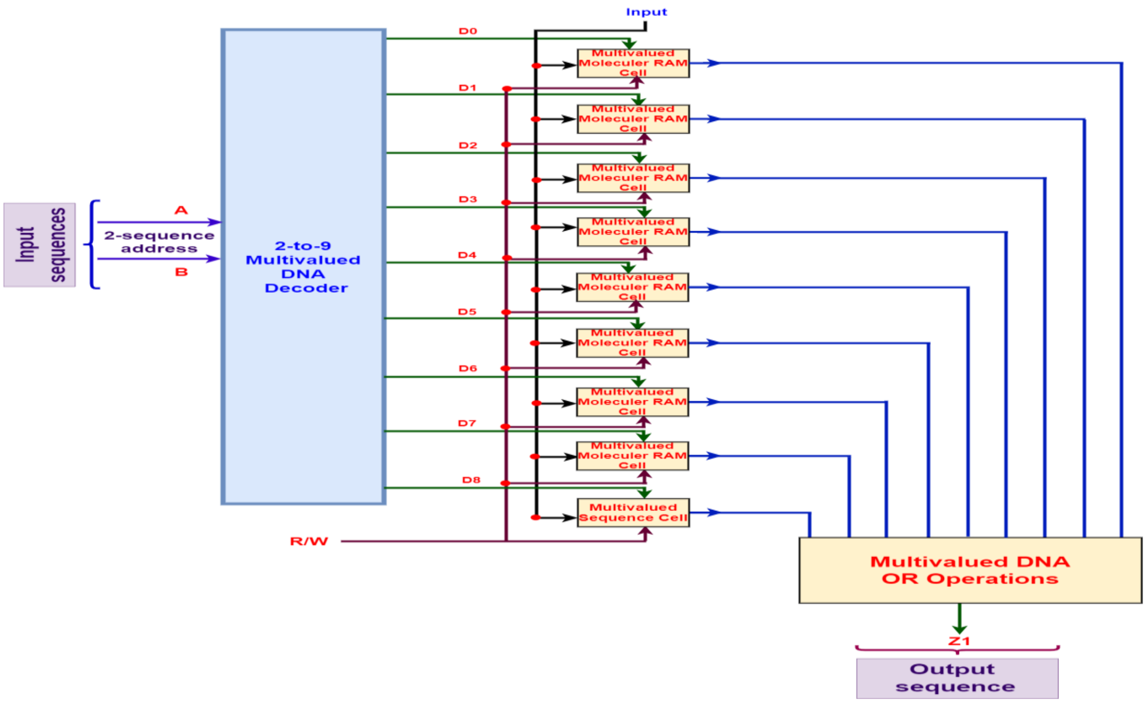
Input
D0
D1
D2
D3
D4
D5
D6
D7
D8
Multivalued Moleculer RAM Cell
Multivalued Moleculer RAM Cell
Multivalued Moleculer RAM Cell
Multivalued Moleculer RAM Cell
Multivalued Moleculer RAM Cell
Multivalued Moleculer RAM Cell
Multivalued Moleculer RAM Cell
Multivalued Moleculer RAM Cell
Multivalued Sequence Cell
Input sequences
A
2-sequence address
B
2-to-9 Multivalued DNA Decoder
R/W
Multivalued DNA OR Operations
Z1
Output sequence

Figure 3: General Organization of 9:1 MSDRAM Circuit Diagram

Step 1: Define the three input sequences: Input, R/W, and Select. Each sequence can exist in one of two states: "TGGATC" (false) and "ACCTAG" (true).

Step 2: Perform the DNA NOT operation on the Input and R/W sequences.

Step 3: Use DNA AND gates, each with three inputs. First, connect the NOT of Input and Select to an AND gate. The output of this AND gate is then connected to another AND gate along with the R/W input sequence.

Step 4: Repeat the process by connecting Input and Select to another AND gate. The output from this AND gate is passed to another AND gate with the R/W input sequence.

Step 5: The outputs from Step 3 and Step 4 serve as inputs to the R-S flip-flop.

Step 6: Finally, the output of the R-S flip-flop and Select is passed through a DNA AND gate. The resulting output, along with the NOT of R/W, is processed through another DNA AND gate to produce the final molecular cell output.

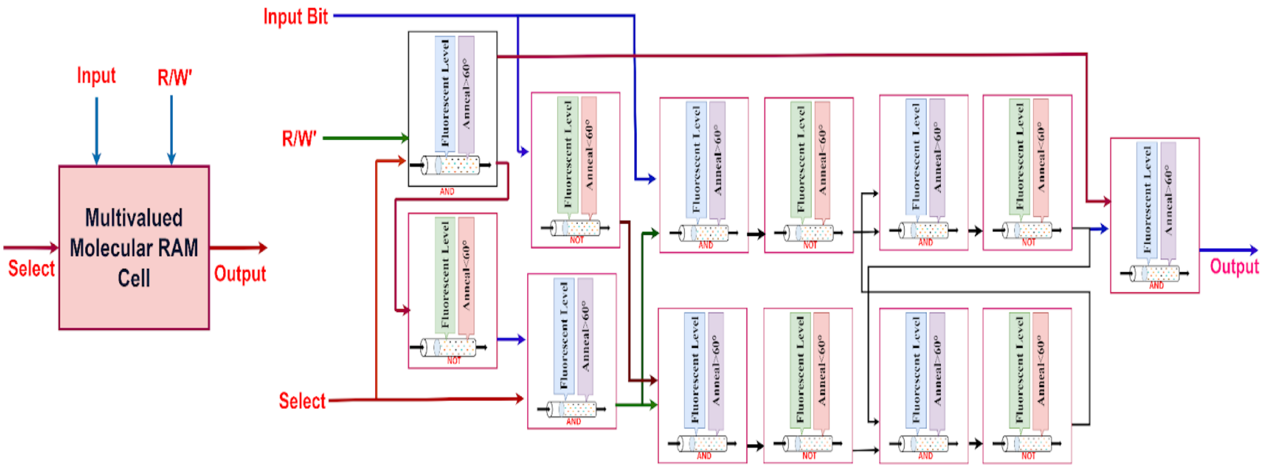

Figure 4: DNA Single Molecular RAM Cell

- ***Working Procedure of Molecular Cells***

One bit of data can be stored in sequential devices using a fundamental R-S flip-flop. To construct a complete memory unit, known as a sequence cell, the flip-flop serves as the foundation. The Select input plays a crucial role in accessing the sequence cell for either reading or writing operations. When the Select input is high, represented by the sequence “ACCTAG,” the memory cell performs its designated operation. Conversely, when the Select input is low, indicated by the sequence “TGGATC,” the memory cell remains inactive and does not perform any read or write operation.

The input sequence “R/W” is controlled by a system clock, where the clock value determines the operation mode. If the read/write line holds the sequence “TGGATC,” it signifies the read operation, and if it holds “ACCTAG,” it performs the write operation. When the cell is selected and in read mode, the current value stored in its flip-flop is transferred to the cell’s output line. Conversely, when the cell is in write mode, the input data signal determines the value to be stored in the flip-flop. Consider a scenario where the cell is selected. If the clock value is

"TGGATC," the cell contents are read, and the output depends solely on the value Input of the flip-flop. If Input is low, the output is "TGGATC," and if high, the output is "ACCTAG." This behavior is achieved because the DNA AND gate connected to the output has three inputs: negated R/W, Select, and Input.

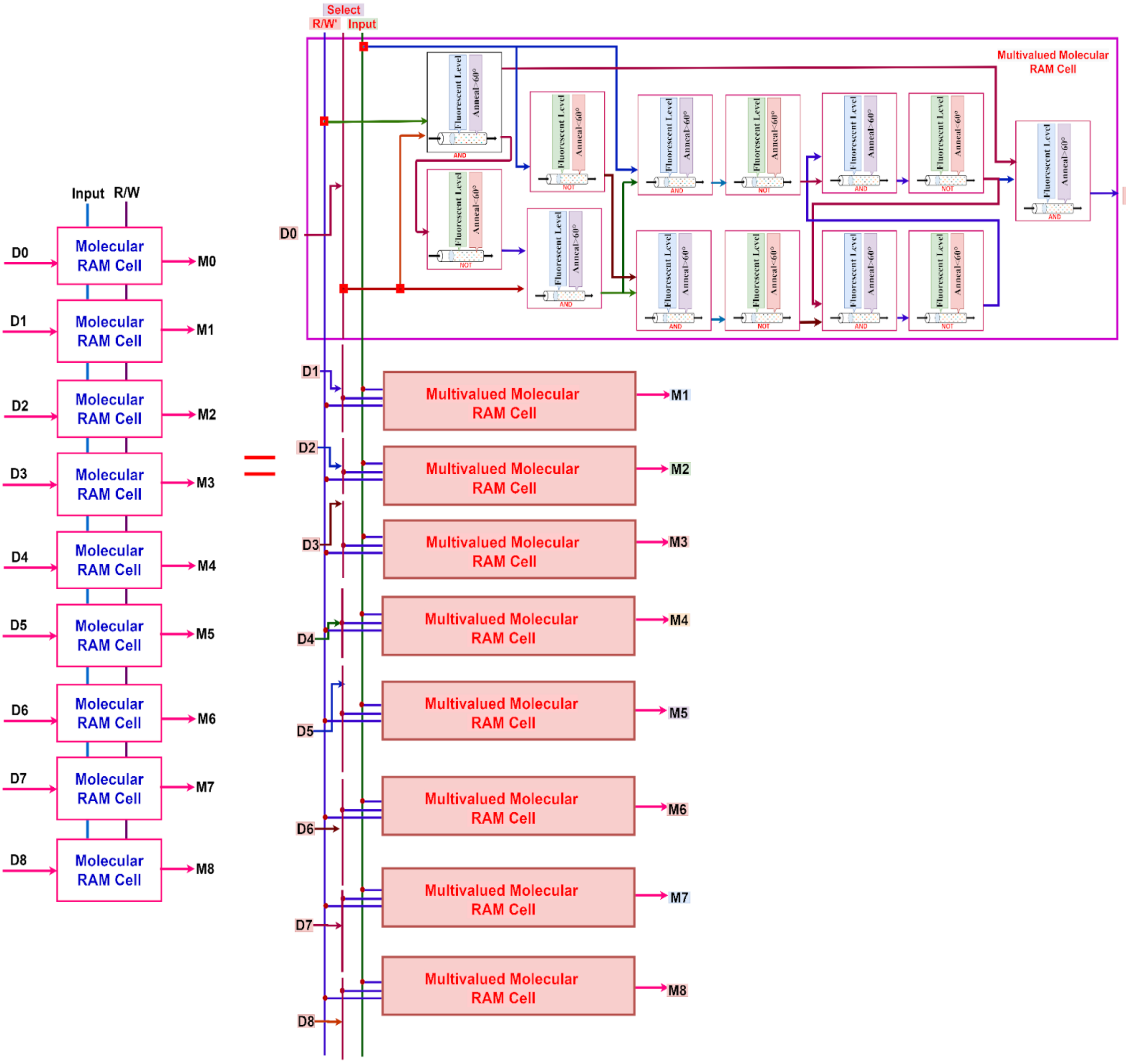

Figure 5: DNA Multivalued Molecular RAM Cell

Here, both negated R/W and Select are high ("ACCTAG"), influencing the output state. Nine DNA sequence cells must be controlled by four selection lines to implement a DNA 9-to-1 RAM. These sequence cells use the output of the DNA 2-to-9 decoder, which produces nine different sequences, as the selection input. Nine sequence cells designated M0 through M8, are shown in Fig 5 and are utilized to carry out additional minterm operations.

3.1.2 Multivalued DNA OR Operation

In a Ternary DNA OR Gate, two input sequences are used to produce a single output. The output is determined by selecting the input sequence with the higher logical value. If both input sequences have the same logical value, one of them is chosen as the output. The detection of sequences and their corresponding logical values is achieved using fluorescent levels.

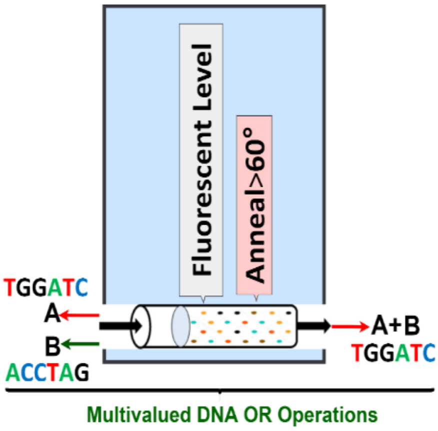

Figure 6: DNA OR Operation

The truth table for the Ternary DNA OR Gate is presented in Table I, and its structural representation is shown in Fig 6.

Table 1: DNA TERNARY OR OPERATION

A OR B		B		
		ACCTAG	CAAGCT	TGGATC
A	ACCTAG	ACCTAG	CAAGCT	TGGATC
	CAAGCT	CAAGCT	CAAGCT	TGGATC
	TGGATC	TGGATC	TGGATC	TGGATC

The DNA decoder output sequences will be transmitted through molecular cells for selection to execute output Z1 of DNA 9-to-1 RAM. Sequences to operate in the sum of minterms form, Z1 = ∑ (M0, M1, M2, M3, M4, M5, M6, M7, M8).

4. Proposed Design of MSDRAM

In the DNA 9-to-1 RAM architecture (Fig 7), two address lines are required, where each address line is in DNA NOT form. These address line combinations serve as inputs to a DNA 2-to-9 decoder, which consists of nine DNA AND gates and an enable input. The decoder generates four select lines, with each line connected to a corresponding molecular cell. The decoder's complexity is defined as k times 3^k. This 9-to-1 sequence RAM is composed of nine distinct molecular cells, where each cell has three inputs: data, a select line, and the read/write input. The outputs of the nine DNA molecular cells are fed into a DNA OR gate, which generates the final output. This completes the design procedure for the DNA 9-to-1 sequence RAM.

- **Working Principle of Multiple-Valued QPROM**

Table II illustrates the functional procedure of the proposed 9-to-1 ternary DNA RAM architecture. The complete circuit of a DNA RAM cell is explained in Fig 4. A word is made up of two DNA RAM cells, designed to allow simultaneous access to both sequences. To access four words of memory, two address lines are required. The address lines A and B serve as inputs to a DNA 2-to-9 decoder, which selects one of the nine available words. The decoder is enabled through the memory-enable input. If the memory-enable signal is TGGATC, all outputs of the decoder will also be TGGATC, ensuring that no memory address is selected. Conversely, when the memory-enable input is ACCTAG, the decoder selects one of the nine words based on the values of the two address lines. Once a word is selected, the read/write input determines the operation. During a read operation, the sequences of the selected word pass through the DNA OR gates to the output terminals Z1. During a write operation, the input data is transferred into the nine DNA cells corresponding to the selected word.

Table 2: CONTROL INPUT TO MEMORY CHIP

R/W	**Memory Operation**
X	None
TGGATC	***Write to selected word***
ACCTAG	***Read from selected word***

In the 9-to-1 DNA RAM, the previously stored sequences remain unchanged. When the memory-enable input to the decoder is TGGATC, none of the words are selected, ensuring that all DNA cells retain their current state regardless of the read/write input value. This describes the operational procedure of the 9-to-1 DNA RAM.

The algorithm of the proposed DNA-based multivalued sequence RAM is illustrated in Algorithm 1.

Algorithm 1: *DNA-based Multivalued Random Access Memory (MSDRAM)*

Input: A, B, Input, R/W;

Output: Z1;

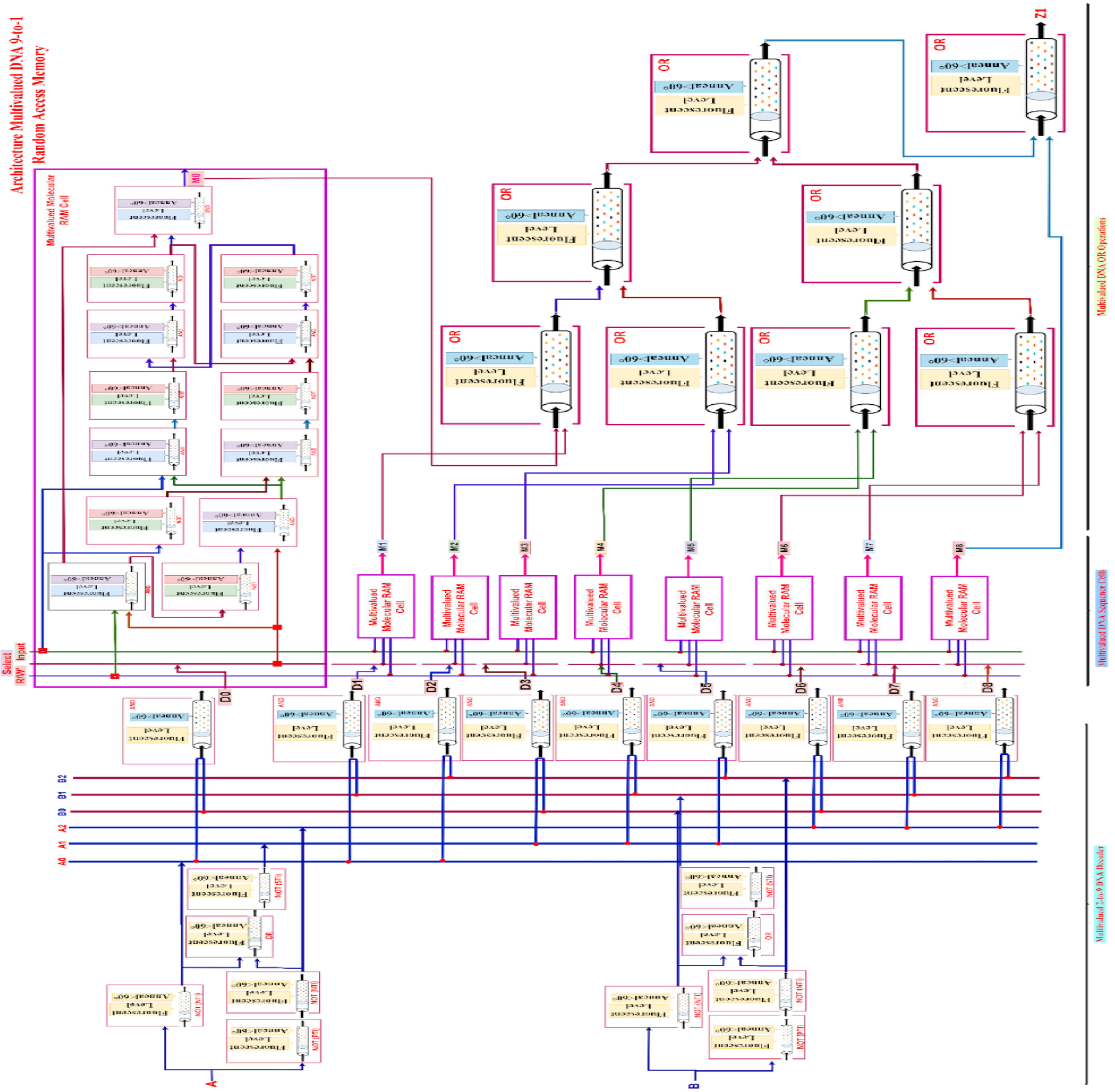
Architecture Multivalued DNA 9-to-1 Random Access Memory
Multivalued Molecular RAM Cell
M0
OR
Anneal>60°
Fluorescent Level
Z1
AND
NOT
Multivalued DNA OR Operations
Multivalued DNA Sequence Cells
Multivalued 2-to-9 DNA Decoder
Multivalued Molecular RAM Cell
M1
M2
M3
M4
M5
M6
M7
M8
Select
R/W
Input
D0
D1
D2
D3
D4
D5
D6
D7
D8
B2
B1
B0
A2
A1
A0
NOT (NT1)
NOT (PT1)
NOT (ST1)
A
B

Figure 7: Multivalued DNA-based 9:1 RAM

The value of A, B, Input, R/W, and Z1 can be ***TGGATC or ACCTAG or CAAGCT***

```
Begin
Procedure DO_DNA_RAM( A, B, Input, R/W )
A0' <- DO_DNA_NOT( A );
```

```
D2 <- DO_DNA_AND( A0' , B );
D3 <- DO_DNA_AND( A, B );
R0 <- DO_DNA_ Molecular_Cell(Input, D0, R/W );
R1<- DO_ DNA_Molecular_Cell (Input, D1, R/W );
R2 <- DO_DNA_Molecular_Cell (Input, D2, R/W );
R3 <- DO_DNA_Molecular_Cell(Input, D3, R/W );
R4 <- DO_DNA_ Molecular_Cell(Input, D4, R/W );
R5<- DO_ DNA_ Molecular_Cell(Input, D5, R/W );
R6 <- DO_DNA_ Molecular_Cell(Input, D6, R/W );
R7 <- DO_DNA_Molecular_Cell (Input, D7, R/W );
R8 <- DO_DNA_Molecular_Cell (Input, D8, R/W );
 B0 <- DO_DNA_OR( R0, R1 );
 B1 <- DO_DNA_OR( R2, R3 );
 B2 <- DO_DNA_OR( R4, R5 );
 B3 <- DO_DNA_OR( R8, R7 );
 F <- DO_DNA_OR( B0, B1 );
F1 <- DO_DNA_OR( B2, B3 );
 F <- DO_DNA_OR( F, F1 );
 Z1 <- DO_DNA_OR( F, R8 );
end procedure
 Procedure DO_DNA_Molecular_Cell( I, S, R/W )
I'  <- DO_DNA_NOT( I  );
A  <- DO_ DNA_AND( S ,  I  );
 S  <- DO_ DNA_AND( A ,  R/W  );
 B <- DO_ DNA_AND( S, R/W  );
 R <- DO_ DNA_AND( B ,  I'  );
 M <- DO_ DNA_OR( S ,  Q' );
 M' <- DO_ DNA_NOT(  M  );
 Q <- DO_ DNA_OR( R ,  M'  );
 Q' <- DO_ DNA_NOT( Q  );
 C <- DO_ DNA_AND(  S ,  Q'  );
 R/W' <- DO_ DNA_NOT(  R/W );
 Y <- DO_ DNA_AND( C,  R/W'  );
 end procedure
Procedure DO_DNA_Multivalued_OR( M, N )
X = fluorescent_get_value(M);
Y= fluorescent_get_value(N);
if X=Y
Do_Result(X);
else if X>Y
```

```
Do_Result(X);
else
Do_Result(Y);
end if
end Procedure
Procedure Do_Result(Q)
if Q=0
return ACCTAG;
else if Q=1
return CAAGCT;
else if Q=2
return TGGATC;
end if
end procedure
End
```

5. Analysis of MSDRAM

5.1. Heat Analysis

A DNA 9-to-1 RAM is a combinational circuit that performs storage as volatile. Input sequences are as follows:

A = **TGGATC**

B = **ACCTAG**

Calculating the melting temperature of a specific DNA sequence:

For A = **TGGATC**

$$T_{m(A)} = 2(A + T) + 4(C + G) - 7$$

$$= 2(1 + 2) + 4(1 + 2) - 7$$

$$= 11.0\ °C$$

Again, Input B = **ACCTAG**

So, $T_{m(\ B)} = 2(A + T) + 4(C + G) - 7$

$$= 2(2 + 1) + 4(2 + 1) - 7$$

$$= 11.0\ °C$$

Specific steps with heat for DNA 9-to-1 RAM (for each tube):

1. Basic gate operation preparing (98 ºC - 94 ºC)
2. Synthesising (98 ºC - 94 ºC)
3. Mixing (95 ºC - 22 ºC)
4. Annealing (70 ºC - 20 ºC)
5. Melting (Depends on the sequence)
6. Amplifying
7. Separating
8. Extracting
9. Cutting (70 ºC - 20 ºC)
10. Ligating
11. Substituting
12. Marking
13. Destroying
14. Detecting and Reading (98 ºC - 25 ºC)

So, in DNA 9-to-1 RAM overall maximum required heat

= (98+98+95+70+11+20+98) ºC

= 490 ºC,

and minimum required heat = (94+94+22+20+11+20+25) ºC = 286 ºC.

In the operation of basic DNA gates, all processes occur within a test tube after the mixing is completed. Depending on the specific steps, the temperature needs to be carefully controlled. For certain stages, the temperature is maintained between 94°C and 98°C, while during DNA gate logic operations, the temperature is adjusted to approximately 20°C. At the detection stage, the temperature is set to 25°C

5.2. Required Time Analysis

DNA-based operations require significant time to execute. To determine the execution time of DNA 9-to-1 RAM, operations are divided into two parallel pipelines to optimize efficiency:

Pipeline 1: NOT, AND, AND, AND, NOR, NOR, AND, AND, OR, OR, OR, OR

Pipeline 2: AND, AND, AND, AND, NOR, NOR, AND, AND, OR, OR, OR, OR

Each basic DNA gate operation (AND, OR, NOT, XOR) requires approximately 2 hours. Additionally, DNA gate preparation takes 6 hours, and fluorescence detection, which is constant for any multi-gate operation, requires 2 hours. Since the second pipeline is the longest, it determines the total processing time. Other operations within the first pipeline will be executed in parallel during this time. The total execution time for DNA RAM is the sum of gate preparation time, gate operational time, and fluorescence detection time. Total Time = Basic Gate Preparement time + Basic Gate Operational Time + Fluorescence Detection Time = (6 + 24 + 2) = 32 hours (approximately). Therefore, the total time required for executing the DNA 9-to-1 RAM is approximately 32 hours.

6. Conclusion

This paper introduces MSDRAM (Multivalued Sequence Storage of Random Access Memory), a pioneering architecture that utilizes the unique properties of DNA technology to address the limitations of conventional silicon-based storage systems. This study presents MSDRAM, a groundbreaking approach to memory storage that leverages the quaternary encoding potential of DNA to achieve high-density, energy-efficient, and scalable storage solutions. By representing multivalued data through DNA sequences, MSDRAM overcomes the storage limitations of conventional silicon-based RAM systems, demonstrating remarkable storage. The proposed architecture integrates advanced error-correction techniques and biochemical access mechanisms to ensure data integrity and reliability. MSDRAM serves as a promising foundation for the future of hybrid memory architectures, combining the benefits of DNA technology with conventional memory systems. Its ability to reduce energy consumption while increasing storage capacity addresses critical challenges in meeting the growing global demand for data storage. This research underscores the transformative potential of molecular memory in revolutionizing the storage landscape. Future directions will focus on optimizing DNA synthesis and retrieval processes, reducing costs, retrieval speeds, and achieving seamless integration with existing computing infrastructure.

References

[1]. Borgman, Christine L. Big data, little data, no data: Scholarship in the networked world. MIT press, 2017.

[2]. Dover, Paul M. The information revolution in early modern Europe. Vol. 62. Cambridge University Press, 2021.

[3]. Groeger, Lars, Kyle Bruce, and Iain Rolfe. "Adapt fast or die slowly: Complex adaptive business models at Cisco Systems." Industrial Marketing Management 77 (2019): 102-115.

[4]. DuBravac, Shawn. Digital destiny: How the new age of data will transform the way we work, live, and communicate. Simon and Schuster, 2015.

[5]. Feynman, Richard P. "Simulating physics with computers." In Feynman and computation, pp. 133-153. cRc Press, 2018.

[6]. Mendell, Madeleine, Mél Hogan, and Deb Verhoeven. "Matters (and metaphors) of life and death: How DNA storage doubles back on its promise to the world." The Canadian Geographer/Le Géographe canadien 66, no. 1 (2022): 37-47.

[7]. Mendell, Madeleine Claire. "DNA in the Archive." PhD diss., Doctoral dissertation, New York University, 2021.

[8]. De Silva, Pavani Yashodha, and Gamage Upeksha Ganegoda. "New trends of digital data storage in DNA." BioMed research international 2016, no. 1 (2016): 8072463.

[9]. Doricchi, Andrea, Casey M. Platnich, Andreas Gimpel, Friederikee Horn, Max Earle, German Lanzavecchia,

Aitziber L. Cortajarena et al. "Emerging approaches to DNA data storage: challenges and prospects." ACS nano 16, no. 11 (2022): 17552-17571.

[10]. Sun, Lichun, Jun He, Jing Luo, and David H. Coy. "DNA and the digital data storage." Health Science Journal 13, no. 3 (2019): 1-7.

[11]. Ceze, Luis, Jeff Nivala, and Karin Strauss. "Molecular digital data storage using DNA." Nature Reviews Genetics 20, no. 8 (2019): 456-466.

[12]. Węgleński, Piotr. Ancient DNA and Genetic Engineering. Wydawnictwa Uniwersytetu Warszawskiego, 2020.

[13]. Manodee, Akarsh, Neeraj Kumar Pandey, Amit Kumar Mishra, and Amit Kumar. "DNA Data Storage: A Solution to the Growing Digital Data Dilemma." In 2024 2nd International Conference on Disruptive Technologies (ICDT), pp. 1374-1379. IEEE, 2024.

[14]. Karouia, Fathi, Kianoosh Peyvan, and Andrew Pohorille. "Toward biotechnology in space: High-throughput instruments for in situ biological research beyond Earth." Biotechnology advances 35, no. 7 (2017): 905-932.

[15]. Zolotovsky, Katia. "Guided growth: design and computation of biologically active materials." PhD diss., Massachusetts Institute of Technology, 2017.

[16]. Meiser, Linda C., Philipp L. Antkowiak, Julian Koch, Weida D. Chen, A. Xavier Kohll, Wendelin J. Stark, Reinhard Heckel, and Robert N. Grass. "Reading and writing digital data in DNA." Nature protocols 15, no. 1 (2020): 86-101.

[17]. Godard, Béatrice, Jörg Schmidtke, Jean-Jacques Cassiman, and Ségolene Aymé. "Data storage and DNA banking for biomedical research: informed consent, confidentiality, quality issues, ownership, return of benefits. A professional perspective." European journal of human genetics 11, no. 2 (2003): S88-S122.

[18]. Sheth, Ravi U., and Harris H. Wang. "DNA-based memory devices for recording cellular events." Nature Reviews Genetics 19, no. 11 (2018): 718-732.

[19]. Akhtar, Zarif Bin, and Ahmed Tajbiul Rawol. "Unraveling the Promise of DNA Data Storage: An Investigative Analysis of Advancements, Challenges, Future Directions." Journal of Information Sciences 23, no. 1 (2024): 23-44.

[20]. Patel, Radhika, Dweepna Garg, Milind Shah, Safeya Dharmajwala, Kush Jindal, and Amit Nayak. "DNA Archives: Revolutionizing Data Storage." In 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), pp. 1270-1275. IEEE, 2023.

[21]. Limbachiya, Dixita, Manish K. Gupta, and Vaneet Aggarwal. "10 years of natural data storage." IEEE Transactions on Molecular, Biological and Multi-Scale Communications 8, no. 4 (2022): 263-275.

[22]. Tabassum, Tamanna, Fatema Akter, and Mohammed Nasir Uddin. "DROM: read only memory storage in DNA computing." In 2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI), pp. 1-6. IEEE, 2022.

[23]. Akter, Fatema, Tamanna Tabassum, and Mohammed Nasir Uddin. "Field Programmable Gate Array in DNA Computing." In International Conference on Big Data, IoT and Machine Learning, pp. 95-107. Singapore: Springer Nature Singapore, 2023.

[24]. Fan, Daoqing, Juan Wang, Erkang Wang, and Shaojun Dong. "Propelling DNA computing with materials’ power: Recent advancements in innovative DNA logic computing systems and smart bio- applications."

Advanced Science 7, no. 24 (2020): 2001766.

[25]. Betz, Ulrich AK. Science and Religion United: The Salvation Machine. Springer Nature, 2024.

[26]. Conrad, Michael. "Cross-scale information processing in evolution, development and intelligence." BioSystems 38, no. 2-3 (1996): 97-109.

[27]. Baiardi, Alberto, Matthias Christandl, and Markus Reiher. "Quantum computing for molecular biology." ChemBioChem 24, no. 13 (2023): e202300120.

[28]. Kozlíková, Barbora, Michael Krone, Martin Falk, Norbert Lindow, Marc Baaden, Daniel Baum, Ivan Viola, Julius Parulek, and H- C. Hege. "Visualization of biomolecular structures: State of the art revisited." In Computer Graphics Forum, vol. 36, no. 8, pp. 178-204. 2017.

[29]. Outeiral, Carlos, Martin Strahm, Jiye Shi, Garrett M. Morris, Simon C. Benjamin, and Charlotte M. Deane. "The prospects of quantum computing in computational molecular biology." Wiley Interdisciplinary Reviews: Computational Molecular Science 11, no. 1 (2021): e1481.

[30]. Kundu, Aniruddha, Sudipta Nandi, and Arun K. Nandi. "Nucleic acid based polymer and nanoparticle conjugates: Synthesis, properties and applications." Progress in Materials Science 88 (2017): 136-185.