Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/20202
DOI URL: http://dx.doi.org/10.21474/IJAR01/20202

RESEARCH ARTICLE

DECOHERENCE-RESILIENT READONLY MEMORY ARCHITECTURE USING QUANTUM TECHNOLOGY

Tamanna Tabassum
Lecturer, Department of Computer Science & Engineering, Ahsanullah University of Science & Technology, Dhaka, Bangladesh.

Manuscript Info

Manuscript History
Received: 08 November 2024
Final Accepted: 14 December 2024
Published: January 2025

Key words:-
Quantum, Qubit, Read Only Memory Architecture, QROM, Superposition

Abstract

Quantum computation draws from theoretical physics, functional analysis, and algorithmic computer science, acting as an interdisciplinary field. The main aim of research in quantum computing is to demonstrate that certain tasks can be completed faster using quantum computers than traditional ones. For this, quantum memory is crucial to developing synchronization tools that coordinate various processes in quantum computers and quantum gates that preserve the identity of quantum states, such as superpositions, and methods to convert preset photons into on-demand photons. Quantum memories are essential for large-scale photonic quantum computing systems, enabling the coherent manipulation, buffering, and retiming of photonic signals. Unlike classical memory, quantum memory allows states to exist in quantum superposition, offering greater flexibility for quantum algorithms than conventional storage systems. While traditional Read-Only Memory (ROM) tends to be slower, quantum computing facilitates the development of novel computer types that operate with qubits as input states, leading to increased storage capacity. This paper proposes a quantum-based ROM (QROM) architecture that utilizes quantum-based binary logic operations and presents an analysis of the system's performance, focusing on heat generation and speed parameters.

Introduction:-

In the past decade, there has been a growing interest among scientists in exploring the interaction between quantum radiation and various particles [1]. One such area of study is quantum memory, which involves transferring the quantum state of light onto a collection of atoms and later retrieving it in its original form. Quantum memory plays an essential role in information processing applications such as optical quantum computing and quantum communication, and it facilitates light-atom interactions [2]. Quantum refers to the smallest possible unit of any physical quantity, such as energy or matter [3]. In classical computing, binary logic is the foundation, where information is represented as 0s and 1s. In contrast, quantum computing relies on qubits for computation [4]. This enables the simulation of quantum systems much faster than conventional computers [5]. The development of quantum gates, or reversible gates, is particularly promising because quantum mechanics itself is inherently reversible. Since the advent of quantum computation, it has been recognized that certain quantum algorithms [6]

Corresponding Author:-Tamanna Tabassum
Address**:-**Lecturer, Department of Computer Science & Engineering, Ahsanullah University of Science & Technology, Dhaka, Bangladesh.

perform much faster than their classical counterparts, and these algorithms require quantum storage and quantum gates for efficient execution.

Read Only Memory (ROM) in the context of quantum computing refers to memory chips that store permanent or semi-permanent data, integrating both quantum decoders and quantum OR operations into a single integrated circuit (IC) [7]. To update the programming in quantum ROM, the chips must be physically removed and replaced, as the data stored in quantum ROM is fixed once the memory device is manufactured [8]. This paper proposes a quantum ROM architecture and analyzes its performance using parameters such as heat and speed in the context of quantum computing.

The paper is organized as follows: Section 2 provides an overview of quantum computing, basic quantum gates, and the structure of ROM. Section 3 introduces the proposed quantum ROM circuit and presents an algorithm to construct ROM logic blocks using quantum computing. The analysis of the proposed quantum ROM based on heat and speed is covered in Section 4. Finally, Section 5 concludes the paper.

Background:-

Traditional computing systems rely on transistors to process data, where each transistor represents a binary state of either 0 or 1. In contrast, quantum computers use qubits, which can exist in superposition, meaning they can simultaneously represent both |0> and |1> states power of quantum systems grows exponentially as more qubits are entangled [9, 10]. Photon qubits, based on quantum exchange, can be stored in quantum memory. Optical data storage is achieved by capturing light frequencies using absorbers, which are then directed to and stored at specific beam space locations. Many early stored-program computers, such as the ENIAC, utilized read-only memory (ROM) for non-volatile program storage [11, 12]. ROM simply required the ability to read data rather than modify it and could be implemented with basic electromechanical devices. In a ROM memory cell, fewer transistors are typically used, with a single transistor connecting a bit line to a word line, indicating either a logical qubit |1> or |0> depending on its presence or absence [13]. Quantum ROM stores the necessary communication between various hardware components. A block diagram of ROM, as shown in Fig. 1, consists of n input lines and m output lines. Each combination of input bits is termed an address, while each combination of output bits is referred to as a word. The number of bits per word is determined by the number of output lines (m) [14, 15]. An address represents a binary number that comes in terms of the n input variables.

Fig. 1:- 2^n-to-m ROM Block Diagram.

The quantum Read-Only Memory (ROM) is initially designed as a combinational circuit, incorporating quantum AND gates functioning as a quantum decoder, along with a number of quantum OR gates corresponding to the output lines of the unit. As a result, it represents a two-level implementation in the sum of minterms form. For a ROM with n input lines and m output lines, the output functions are calculated in the sum of the minterms form. The number of distinct addresses possible with n input variables is 2^n. Each output word is selected by a unique address, meaning that with 2^n distinct addresses, the ROM can store 2^n distinct words. The word that appears on the output lines at any given time depends on the address value applied to the input lines. Therefore, a ROM is defined by the number of words (2^n) and the number of bits per word (m). For example, if there are 2 input lines (n=2) and 2 output lines (m=2), the ROM is referred to as a 4-to-2 ROM, and its output functions are expressed as F1 and F2 in the sum of minterms form, $\sum(0, 1, 2, 3)$ [16].

A quantum gate, a fundamental component of quantum circuits, operates on multiple qubits. In the realm of quantum computing, reversible logic circuits are systematized using three primary gates: the controlled-NOT (CNOT) gate, the controlled V gate, and the controlled V+ gate [17].

Proposed System Architecture:-

The quantum 4-to-2 Read-Only Memory (ROM) block diagram (referenced as Fig. 2) comprises four words, each consisting of two input qubits (|A⟩ and |B⟩). This design results in two output qubits (|F_1⟩ and |F_2⟩) and four distinct stored words. The output lines display a specific word selected based on the combination of the two input qubits. Since the input qubits can represent four unique states ($2^2 = 4$), the ROM requires four addresses. Implementing the minterms for these addresses involves the use of a Quantum 2-to-4 decoder in conjunction with Quantum OR operations.

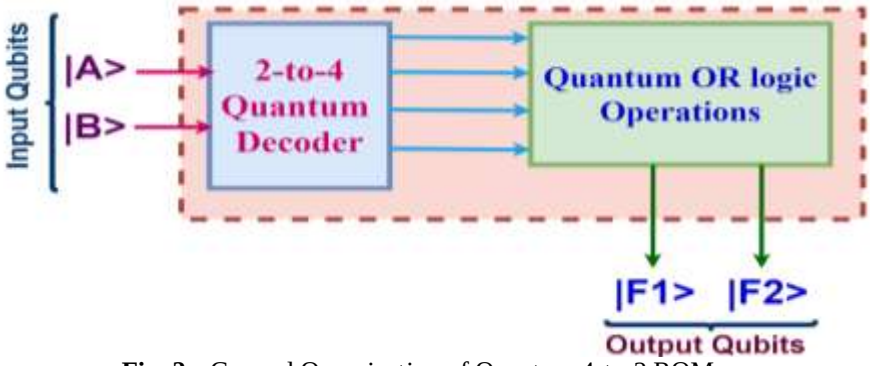

Fig. 2:- General Organization of Quantum 4-to-2 ROM.

Each input qubit address corresponds to a unique word selection. For instance, when the input qubit address is |0⟩|0⟩, the system selects the word 0, which is then displayed on the output qubit lines. Similarly, if the input qubit address is |1⟩|1⟩, the system selects word 3, and this word is presented on the output lines.

Architecture Of Basic Components:-

Quantum Decoder And Or Operations:-

A quantum 2-to-4 decoder is a combinational logic circuit constructed using two quantum NOT gates and four quantum AND gates (refer to Fig. 3). Initially, quantum NOT gates are connected to the input qubits |A⟩ and |B⟩. The first output qubit, |D_0⟩ = |A'⟩ · |B'⟩, is generated by applying a quantum AND operation to the outputs of these NOT gates. Next, a quantum NOT gate is connected to the input qubit |B⟩, followed by a connection to |A⟩, and the resulting signals undergo a quantum AND operation to produce the second output qubit, |D_1⟩ = |A⟩ · |B'⟩. Similarly, another NOT gate is connected to |A⟩, followed by |B⟩, and their signals are combined with a quantum AND operation to generate the third output qubit, |D_2⟩ = |A'⟩· |B⟩. Finally, the input qubits |A⟩ and |B⟩ directly undergo a quantum AND operation to produce the fourth output qubit, |D_3⟩ = |A⟩ · |B⟩.

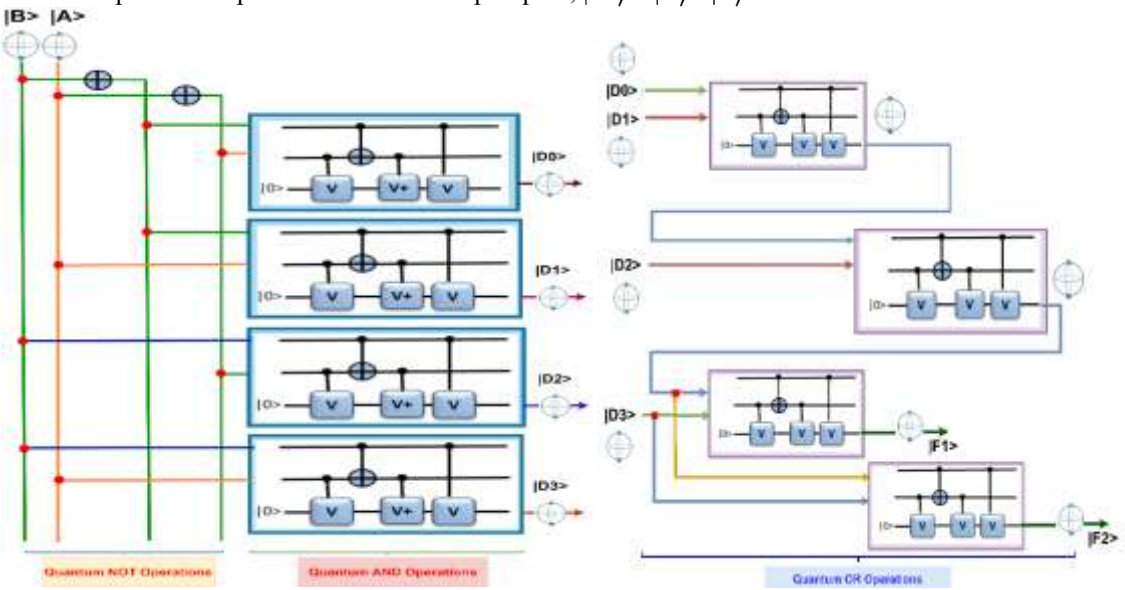

Fig. 3:- Quantum Decoder.**Fig. 4:-** Quantum OR Operation.

The quantum OR operation involves two variable input qubits (used to control the gates) and a constant input qubit |0⟩, which passes through the gates on the target output line to produce the required output. To generate the outputs $|F_1\rangle$ and $|F_2\rangle$ for the Quantum 4-to-2 ROM, the decoder's output qubits are combined in the sum-of-minterms form: $|F_1\rangle, |F_2\rangle = \Sigma(0, 1, 2, 3)$. First, $|D_0\rangle$ and $|D_1\rangle$ are processed through a quantum OR operation. The resulting output is then combined with $|D_2\rangle$ using another quantum OR operation. Finally, $|D_3\rangle$ is combined with the previous OR operation output using one more quantum OR gate to produce the final outputs $|F_1\rangle$ and $|F_2\rangle$ (refer to Fig. 4).

The QROM logic block is constructed by integrating a decoder with OR operations. The sum of minterms is utilized to produce the required outputs in the QROM. A binary-valued quantum-based ROM is depicted in Fig. 5. The outputs from the decoder AND gates are passed through OR gates to generate the logical outputs of the ROM. Based on the truth table of the Quantum 4-to-2 ROM (refer to Table 1) and the operations specified therein, the following steps are necessary to achieve the desired output qubits:

[i] For input qubits |A⟩, |B⟩ = |0⟩, |0⟩, the $|D_0\rangle$ line is activated. In this case, $|D_0\rangle = |1\rangle$, while $|D_1\rangle$, $|D_2\rangle$, and $|D_3\rangle$ are set to |0⟩. The outputs $|F_1\rangle$ and $|F_2\rangle$ are generated by performing OR operations on $|D_0\rangle = |1\rangle$, $|D_1\rangle = |0\rangle$, $|D_2\rangle = |0\rangle$, and $|D_3\rangle = |0\rangle$, resulting in an output of |1⟩.

[ii] For input qubits |A⟩, |B⟩ = |1⟩, |0⟩, the $|D_1\rangle$ line is activated. Here, $|D_1\rangle = |1\rangle$, while $|D_0\rangle$, $|D_2\rangle$, and $|D_3\rangle$ are set to |0⟩. The outputs $|F_1\rangle$ and $|F_2\rangle$ are computed by performing OR operations on $|D_0\rangle = |0\rangle$, $|D_1\rangle = |1\rangle$, $|D_2\rangle = |0\rangle$, and $|D_3\rangle = |0\rangle$, producing an output of |1⟩.

[iii] For input qubits |A⟩, |B⟩ = |0⟩, |1⟩, the $|D_2\rangle$ line is activated. In this scenario, $|D_2\rangle = |1\rangle$, while $|D_0\rangle$, $|D_1\rangle$, and $|D_3\rangle$ are set to |0⟩. The outputs $|F_1\rangle$ and $|F_2\rangle$ are obtained by performing OR operations on $|D_0\rangle = |0\rangle$, $|D_1\rangle = |0\rangle$, $|D_2\rangle = |1\rangle$, and $|D_3\rangle = |0\rangle$, resulting in an output of |1⟩.

[iv] For input qubits |A⟩, |B⟩ = |1⟩, |1⟩, the $|D_3\rangle$ line is activated. In this case, $|D_3\rangle = |1\rangle$, while $|D_0\rangle$, $|D_1\rangle$, and $|D_2\rangle$ are set to |0⟩. The outputs $|F_1\rangle$ and $|F_2\rangle$ are derived by performing OR operations on $|D_0\rangle = |0\rangle$, $|D_1\rangle = |0\rangle$, $|D_2\rangle = |0\rangle$, and $|D_3\rangle = |1\rangle$, yielding an output of |1⟩.

Table 1:- Truth table of Quantum 2-to-4 ROM Circuit.

|B>	**|A>**	**|F1>**	**|F2>**
|0>	|0>	|1>	|1>
|0>	|1>	|1>	|1>
|1>	|0>	|1>	|1>
|1>	|1>	|1>	|1>

QROMAlgorithm:-

Algorithm 1: Quantum-based Read Only Memory (QROM)

Input: |A>, |B>

Output: |F1>, |F2>;

The value of |A>, |B>, |F1>, |F2> can be |0> or |1>

```
1.    Begin
2.    while i equal to 1 to ndo
3.    |P> = DO_Quant_Decoder( |Ai>, |Bi> ); // Decoder generates |D0> - |D3>
4.    |P0><- DO_Quant_OR( |D0>, |D1> );
5.    |P1><- DO_Quant_OR( |D2>, |P0> );
6.    |F1><- DO_Quant_OR( |D3>, |P1> );
7.    |F2><- DO_Quant_OR( |D3>, |P1> );
8.    end while
9.    Procedure DO_Quant_Decoder( |A>, | B> )
10.    |D0><- DO_Quant_AND( |A'>, |B'> );
11.   |D1><- DO_Quant_AND( |A>, |B'> );
12.   |D2><- DO_Quant_AND( |A'>, |B> );
13.   |D3><- DO_Quant_AND( |A>, |B> );
14.   end Procedure
15.   Procedure DO_Quant_AND( | A >, | B > )
16.     if | A > and | B > both are | 0 >
17.           return | 0 >
18.     else if | A > is | 0 > and | B > is | 1 >
19.           | P ><- PerformVPlusOp( | 1 > );
```

```
20.                 | Q ><- PerformVOp( | P > );
21.                 | R ><- PerformNOTOp( | Q > );
22.           else if | A > is | 1 > and | B > is | 0 >
23.                 | P ><- PerformVOp( | 1 > );
24.                 | Q ><- PerformVPlusOp( | P > );
25.                 | R ><- PerformNOTOp( | Q > );
26.           else if | A > and | B > both are | 1 >
27.                 | P ><- PerformVPlusOp( | 1 > );
28.                 | Q ><- PerformVPlusOp( | P > );
29.                 | R ><- PerformNOTOp( | Q > );
30.           end if
31.       end procedure
32.       Procedure DO_Quant_OR( | A >, | B > )
33.           if | A > and | B > both are | 0 >
34.                 return | 0 >
35.           else if | A > is | 0 > and | B > is | 1 >
36.                 | P ><- PerformVOp( | 1 > );
37.                 | Q ><- PerformVOp( | P > );
38.                 | R ><- PerformNOTOp( | Q > );
39.           else if | A > is | 1 > and | B > is | 0 >
40.              | P ><- PerformVOp( | 1 > );
41.                 | Q ><- PerformVOp( | P > );
42.                 | R ><- PerformNOTOp( | Q > );
43.           else if | A > and | B > both are | 1 >
44.                 | P ><- PerformVOp( | 1 > );
45.                 | Q ><- PerformVOp( | P > );
46.                 | R ><- PerformNOTOp( | Q > );
47.           end if
48.       end procedure
49.       End
```

Analysis of Proposed System Architecture:-

Heat Calculation:-

Quantum 4-to-2 ROM is a 3-qubit quantum operation, and for this quantum operation, heat measurement will be calculated using [18, 19] the following formula (i).

$$\frac{dS_{th}(E)}{dE} = \frac{n}{T} \quad \ldots \quad (i)$$

We know, for N qubit gate, $S_{th}(E, N) = N(K_B \ln 2)\, S(E/N) = n\left(-k_B \frac{\epsilon-\frac{E}{n}}{\epsilon} \ln \frac{\epsilon-\frac{E}{n}}{\epsilon} - k_B \frac{\frac{E}{n}}{\epsilon} \ln \frac{\epsilon\times\frac{E}{n}}{\epsilon}\right)$

If it is 3 qubits, then N = 3. $S_{th}(E)$ is quantum mechanics qubit entropy.

So, $S_{th}(E, N) = 183.6798 \times 10^{-20}$

$T = \frac{dE}{dS_{th}(E)} = \frac{5.134 \times 10^{-16}}{183.6798 \times 10^{-20}} = 279.51$ k

Speed Calculation:-

A ROM is a device that can receive multiple input signals and synthesize singles with specific functional output signals in a recoverable manner for each input signal. To find the required performing time of Quantum 4-to-2 ROM, divide it into two pipelines as some of the basic quantum gate operations are performed in parallel [20, 21]. Two pipelines are as follows: 1. NOT, AND, OR, OR, OR & 2. AND, OR, OR, OR. Performing time for 4-to-2 ROM will be (NOT + AND + OR + OR + OR) μs.

Where the required time for the basic quantum NOT gate is 1 μs, the required time for the basic quantum AND gate is 30 μs, and the required time for the basic quantum OR gate is 30 μs.

Total required time for Quantum 4-to-2 ROM is = (1 + 30 + 30 + 30 + 30) = 121 μs.

Conclusion:-

A ROM uses fuses or anti-fuses to lock the configuration of each bit, ensuring the stored information is permanent and cannot be modified. ROMs are commonly used in digital electronic devices to store low-level programs, such as firmware or microcode, as well as persistent data. Since the data is written into the ROM during manufacturing, these devices are typically used for large-scale production runs with thoroughly validated data. The primary objective of developing quantum-based ROM is to provide cost-effective, robust, high-density, reliable, and energy-efficient memory solutions. Quantum-based ROM technology is designed to resist degradation over time while allowing for rapid data writing, reading, and erasing. This paper presents the design and analysis of a ROM architecture where quantum gates are employed to implement memory operations. The proposed approach has the potential to reduce energy consumption, simplify QROM implementation significantly, and achieve an exponentially lower decoherence rate.

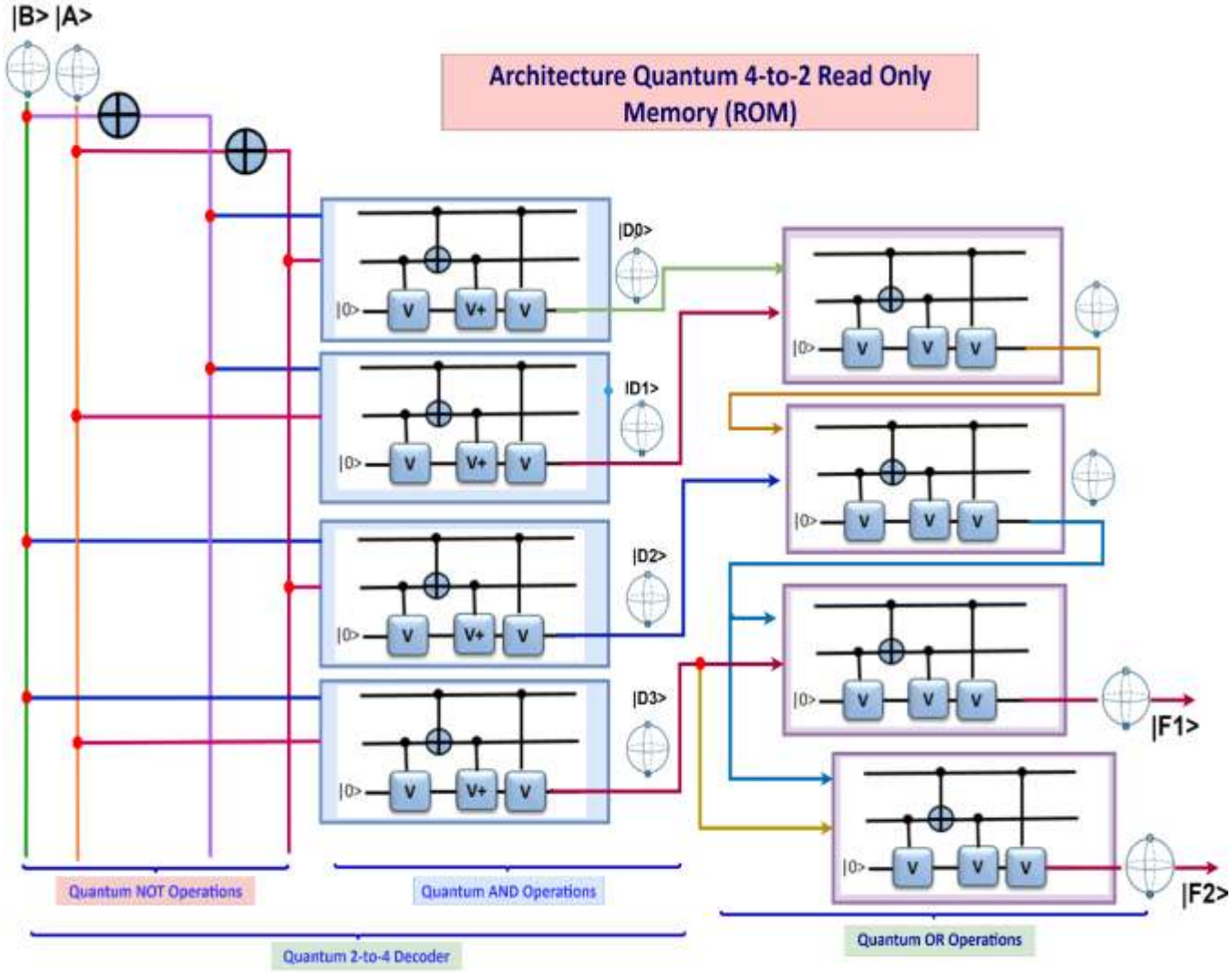

Fig. 5:- Architecture 4-to-2 QROM.

References:-

1. Burton, David A., and Adam Noble. "Aspects of electromagnetic radiation reaction in strong fields." Contemporary Physics 55, no. 2 (2014): 110-121.
2. Bussieres, Félix, Nicolas Sangouard, Mikael Afzelius, Hugues De Riedmatten, Christoph Simon, and Wolfgang Tittel. "Prospective applications of optical quantum memories." Journal of Modern Optics 60, no. 18 (2013): 1519-1537.
3. Koons, Robert C. "Hylomorphic escalation: An Aristotelian interpretation of quantum thermodynamics and chemistry." American Catholic Philosophical Quarterly (2017).

4. De Vries, Andreas. Quantum computation: An introduction for engineers and computer scientists. BoD–Books on Demand, 2012.
5. Taddei, M.M., Escher, B.M., Davidovich, L. and de Matos Filho, R.L., 2013. Quantum speed limit for physical processes. Physical review letters, 110(5), p.050402.
6. Shor, Peter W. "Introduction to quantum algorithms." In Proceedings of Symposia in Applied Mathematics, vol. 58, pp. 143-160. 2002.
7. Stanisavljević, Miloš, Alexandre Schmid, and Yusuf Leblebici. Reliability of nanoscale circuits and systems: methodologies and circuit architectures. Springer Science & Business Media, 2010.
8. Lv, Ziyu, Yan Wang, Jingrui Chen, Junjie Wang, Ye Zhou, and Su-Ting Han. "Semiconductor quantum dots for memories and neuromorphic computing systems." Chemical reviews 120, no. 9 (2020): 3941-4006.
9. Marella, Surya Teja, and Hemanth Sai Kumar Parisa. "Introduction to quantum computing." Quantum Computing and Communications (2020).
10. Singh, Jasmeet, and Mohit Singh. "Evolution in quantum computing." In 2016 International Conference System Modeling & Advancement in Research Trends (SMART), pp. 267-270. IEEE, 2016.
11. Butterfield, Andrew, Gerard EkembeNgondi, and Anne Kerr, eds. A dictionary of computer science. Oxford University Press, 2016.
12. Lo, Hoi-Kwong, Tim Spiller, and Sandu Popescu. Introduction to quantum computation and information. World Scientific, 1998.
13. Gottesman, Daniel. "An introduction to quantum error correction and fault-tolerant quantum computation." In Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics, vol. 68, pp. 13-58. 2010.
14. Tyson, Jeff. "How ROM works." (2004).
15. Cappelletti, Paulo, Carla Golla, Piero Olivo, and Enrico Zanoni, eds. Flash memories. Springer Science & Business Media, 2013.
16. M. Morris Mano, Charles R. Kime, and Tom Martin. "Logic and computer design fundamentals." (2015).
17. Surekha, M. "Efficient approaches for designing quantum costs of various reversible gates." International Journal of Engineering 9, no. 1 (2017): 57-78.
18. Saxena, Surendra K. "Thermodynamics of Epsilon Iron at Core Physical Conditions." Earth's Core: Dynamics, Structure, Rotation 31 (2003): 129-136.
19. Diósi, Lajos. A short course in quantum information theory: an approach from theoretical physics. Vol. 827. Springer, 2011.
20. Levitin, Lev B., Tommaso Toffoli, and Zachary Walton. "Operation time of quantum gates." arXiv preprint quant-ph/0210076 (2002).
21. Thaker, D. D., Metodi, T. S., Cross, A. W., Chuang, I. L., & Chong, F. T. (2006, June). Quantum memory hierarchies: Efficient designs to match available parallelism in quantum computing. In the 33rd International Symposium on Computer Architecture (ISCA'06) (pp. 378-390). IEEE.