

# Optimized Synchronous Dual-Port Memory Design Using Flip-Flop Clock Gating

Matta Yashwanth Reddy<sup>1</sup>; Dr. G. Sujatha<sup>2</sup>

<sup>1</sup>Department of Electronics and Communication Engineering Sri Venkateswara College of Engineering (Autonomous), Tirupati, India

<sup>2</sup>Department of Electronics and Communication Engineering Sri Venkateswara College of Engineering (Autonomous), Tirupati, India

Publication Date: 2025/11/19

**Abstract:** This paper introduces a unique dual-port RAM design that targets both power efficiency and timing performance optimization. The new architecture merges the sophisticated clock gating and the data-transfer pipeline techniques, which make clocks on and off as per the need thus reducing the dynamic power to the most minimum while still using the pipelining technique to its full advantage for throughput and stabilization of timing. In contrast to the classical dual-port RAMs, this design has the simultaneous read and write feature without the energy cost that is traditionally associated with it. The system which is built in Verilog HDL and runs on FPGA platforms overcomes port conflicts by using techniques of port locking and priority management. The results of the tests indicate that there are major savings in power without any degradation in the functional performance, hence this dual-port RAM can be particularly utilized in the applications where efficient parallel memory operation under a strict power constraint is also required. The power-optimized RAM module with robust conflict resolution that is one of the contributions of this work is already showing its potential for high-speed, low-power applications.

**Keywords:** Dual-Port RAM, FPGA, Verilog HDL, Clock Gating, Pipelining, Concurrent Read/Write, Conflict Resolution.

**How to Cite:** Matta Yashwanth Reddy; Dr. G. Sujatha (2025) Optimized Synchronous Dual-Port Memory Design Using Flip-Flop Clock Gating. *International Journal of Innovative Science and Research Technology*, 10(11), 867-876. <https://doi.org/10.38124/ijisrt/25nov504>

## I. INTRODUCTION

The development of memory is crucial for high-performance computing and embedded systems throughout the world since these aspects of design will determine the efficiency, speed, and power consumption of the systems. The dual-port RAM (DPRAM) is one of these memory structures that are vital in the case of applications that necessitate the simultaneous read and write operations, which are some of the mentioned fields like digital signal processing, real-time image processing, communication systems, and FPGA implementations. DPRAM due to its characteristic to allow concurrent memory access via its two ports becomes thus letting the processing environments to maximize throughput and minimize delays in data transfer.

On the other hand, the first generation dual-port RAM designs do not have it easy, they encounter a lot of problems. The major one piles up with the aforementioned dual-port operation that grants performance on one side and increases dynamic power consumption on the other, since both ports are active and even slight idle in use leads to power loss. Moreover, the problem gets obscured with the timing constraints that come along with higher operating

frequencies, especially when pipelining techniques are not properly tuned resulting in the scenario where mobile electronics and edge inference applications that demand both low power and high performance are facing issues in terms of scalability

A number of solutions have been considered to counter such challenges. Sparing power memory structures and hierarchical cell designs of memory cells cut switching activity and enhance energy efficiency. Clock gating has also seen extensive application of minimizing power consumption by disabling parts of the memory that are not under utilization. These methods however should be well balanced to ensure performance is not compromised. Pipelining has the ability to increase throughput by minimising the delay of the critical path and higher operating frequencies are made possible. Nonetheless, embarking on over pipelining can bring about latency and design sophistication.

Conflict handling, which involves two ports trying to access the same memory address at the same time, is another major problem in the design of the dual-port RAM. Such conflicts require efficient arbitration and port-locking

schemes to settle them. Classical priority-based schemes are effective in low-frequency contention however, when it comes to a high number of memory access conflicts then the scheme becomes ineffective resulting in bottlenecks. Although there have been more advanced schemes of arbitration, they tend to involve extra hardware making the whole design less scalable.

In this paper, we introduce a new type of dual-port RAM architecture which is able to cope with these challenges. We propose a design that utilizes both the state of the art clock gating and pipelining methods and is optimized to provide the best power consumption and timing performance. Verilog HDL is used to implement the architecture and it is deployed on the FPGA systems to test its feasibility and scalability. The most significant contributions of this work are the following:

- A two-port RAM architecture which consumes less power by selectively gating clock.
- Further additions to pipeline stages in order to improve timing performance and throughput.
- FPGA implementation and validation- will prove power savings at no loss of dual-port functionality.
- Caching of priority-based conflict resolution and locking of ports, which gives predictable behaviour with contention.

The remaining part of this paper follows the following structure: Section 2 addresses related work and limitations of the current dual-port RAM designs. The section 3 provides the proposed architecture including the clock gating and pipelining strategies. Section 4 shows the experimental findings and compares them with baseline designs. The last section of the paper, Section 5, wraps the paper and gives recommendations on future research.

## II. LITERATURE REVIEW

Reddy et al. [1] introduced a multi-clock dual-port RAM design using Verilog, which allowed read and write can take place (concurrently) on dissimilar ports. The paper has focused on modularity and scalability, which brings the advantages of multi-clock architecture to parallel processing systems. In the same manner, Abumwais et al. [2] presented a dual-port content-addressable memory (CAM) optimized towards cache memory application that exhibited fast access

to data and optimized parallel search functions that are needed in performance-critical systems.

Jang [3] discussed pipelined hardware design, and how hazard mitigation of pipelined circuits can be enhanced to better withstand with throughput without affecting reliability. This idea is essential to designs of DPRAM with timing and data hazards potentially lowering efficiency. In a similar work, Arora et al. [4] designed compute-in-memory (CoMeFa) FPGA blocks, which minimize data movement and allow the definition of energy efficiency, which supports the significance of memory-centric computation in high-performance applications. Marino et al. [5] also maximized FPGA memory access to vision transformers using ME-ViT and showed memory-efficient methods of single-load techniques that reduced energy usage in large models. Recently, Dhakad and Vishvakarma [6] investigated configurable multi-port memory architectures to high-speed data communication and gave a scalable framework that can be used to support several parallel access conditions.

Clock gating and low-power methodologies play the key role in the energy efficient DPRAM design. Al-Zawawi et al. [7] studied low-power clock gating of sequential circuit design and minimized dynamic power without any reduction in performance. A low-power dual-port memory architecture was suggested by Morgenshtein et al. [8] as a high-performance one. SoC applications, demonstrating that careful architectural modifications can simultaneously optimize energy and throughput. Esmaeilzadeh et al. [9] investigated neural acceleration for approximate memory, emphasizing trade-offs between accuracy, energy, and speed, highlighting potential strategies for low-power memory design in AI workloads.

Simulation and modeling frameworks are essential for evaluating DPRAM performance. Kim et al. [10] developed Ramulator, a fast and extensible DRAM simulator for architectural research, useful for verifying memory access patterns and latency impacts. Pan et al. [11] proposed pipelined error control mechanisms for low-power, high-speed on-chip interconnects, aligning with the need for reliable high-frequency memory operation. Abella et al. [12] explored adaptive memory compression for energy and performance optimization, demonstrating the benefits of intelligent memory management techniques to reduce bandwidth and power consumption in high-performance computing.

### III. PROPOSED METHOD

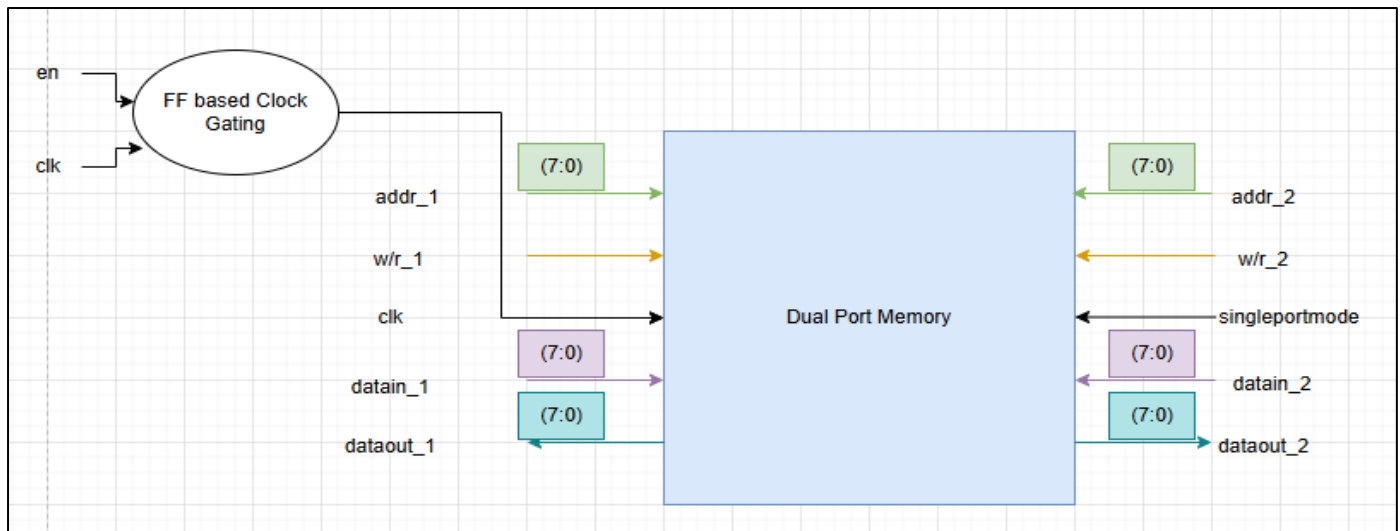


Fig 1 Block Diagram of DP-Memory

This figure illustrates the architecture of a dual-port memory system that incorporates a flip-flop (FF)-based clock gating mechanism. The system consists of the following primary components:

- **Clock Gating Logic:**

The enable signal (en) determines whether the clock signal (clk) is passed or not on the clock gating. At a high en value, the clock is free to pass over onto the flip-flops to allow memory operations.

- **Dual-Port Memory:**

The memory contains two ports which are independent and can read and write information at the same time. The write/read signals (wr, rd) of each port control the data operations. Address Inputs (addr\_1, addr 2) point to the appropriate address to read or write. Data Inputs (datain\_1, datain\_2) are the data being written and Data Outputs (dataout\_1, dataout\_2) are the data being read in the memory.

- **Control Signals:**

There are control signals (wr\_1, wr\_2, rd\_1, rd 2) to support write and read activities. The addresses ( addr 1,

addr 2) identify the memory address whereas the data lines ( datain 1, datain 2, dataout 1, dataout 2 ) refer to data input/output of each port.

- **System Architecture:**

The system permits each port to get access to locations in memory independently. The W/R signals are used to identify whether the operation is the read (0) or write (1). The array consists of 256 locations in memory and both ports can read/write any location in the array. Single Port Mode: This mode is enabled when both ports use one common address and data space thus effectively disabling Port 2. Any access to Port 2 causes an undefined output (8'bx), meaning that it is not active.

- **Operational Overview:**

In addr 1 (Port 1) when the W/R 1 = 1, the data of datain 1 is written into the memory at addr 1. Assuming W/R 1 = 0, the data in addr 1 is read and transmitted out via the dataout 1. Likewise, when W/R\_2 = 1, at addr (Port 2), the data in datain(2) is stored in memory at addr 2, and when W/R 2 = 0, the data in addr 2 is read and sent to dataout (2).

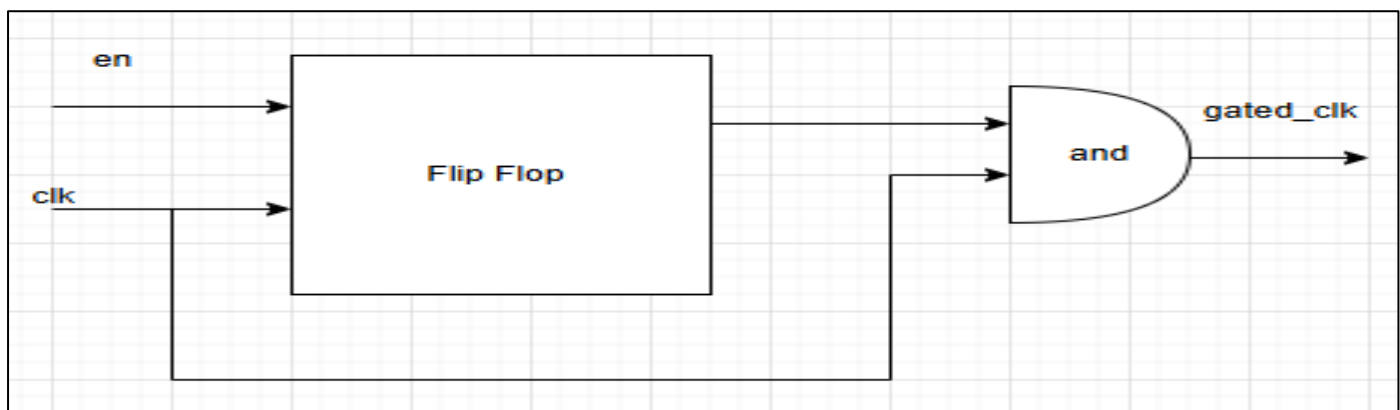


Fig 2 Flipflop Based Clock Gating

This diagram represents a clock gating system based on flip-flop and selectively switches the clock signal to reduce power consumption. These are the following components:

- **Clock Gating Logic:**

An AND gate is fed on the clock signal (clk) and the enable signal (en), it produces the gated clock signal (gated\_clk). The flip-flop stores the data and its state is altered according to the gated clock signal.

- **Operation:**

The logic signal of the clock reaches the flip-flop through the AND gate when the en signal is high (logic 1). The clock signal is then used to update the state of the flip-flop. In case of low en (logic 0) the clock signal is blocked to block extraneous clock cycles and minimise power use by removing idle states in the flip-flop.

- **Pipelined Data Processing:**

Pipelined stages are used to control the memory operations in the system: Stage 1: The data inputs and address are enabled on the rise of the gated clock. Stage 2: Memory operations (write or read) takes place with respect to the write enable signal. Stage 3: Memory data is passed to memory to the respective registers.

**Conflict Resolution:** Where both ports are trying to access the same memory location, Port 1 is a priority. The two ports will attempt to write towards the same address, the data of Port 2 is not written, this will guarantee that it will operate reasonably. The above figures depict the design and working of a dual port memory system with clock gating and pipelining which is designed to save power and provide a memory access which is efficient. Fig. 1 is a description of the general memory format, clock gating system, dual-port memory format, and data read/write control signals. Fig. 2 describes the clock gating system based on flip-flops, which selectively gate the clock signal in order to reduce unnecessary power consumption and also in order to make sure that the memory operates effectively.

These graphics are very clear on how the system functions in parallel, conflict management, and the use of pipelined stages that had optimal performance at minimal power usage. Clock gating is used to ensure that only the power is used when needed and pipelining is used to enable the system to operate at more high speeds without causing too much latency.

Table 1 Operations on Port 1 and Port 2

| Port 1 | Port 2 | Condition                                    | Operation                                                                   |
|--------|--------|----------------------------------------------|-----------------------------------------------------------------------------|
| Write  | Read   | Address1=Address2<br>Or<br>Address1≠Address2 | Write data from Port 1 to Address 1 and read from Address 2 through port 2  |
| Read   | Write  | Address1=Address2<br>Or<br>Address1≠Address2 | Write data from Port 2 to Address 2 and read from Address 1 through port 1  |
| Read   | Read   | Address1=Address2Or<br>Address1≠Address2     | Read from Address 1 through Port 1 and read from Address 2 through Port 2   |
| Write  | Write  | Address1≠Address2                            | Write data from Port 1 to Address 1 and write data from Port 2 to Address 2 |
| Write  | Write  | Address1=Address2                            | Write data from Port 1 to Address 1 and drop data at Port 2                 |

Table 1 tells how the dual-port memory system works depending on the condition of Port 1 or Port 2 with regard to its read and write operations. Where the two ports are going to the same address (Address1 = Address2) then the data is written to the address using Port 1 using Port 2. When one port is reading and the other port is writing to different addresses (Address1 ≠ Address2) there is a possibility of data being written to Address2 by Port 2 and data being written to Address1 by Port 1. In the cases when ports are programmed to read to alternate addresses (Address1 ≠ Address2) data is read to both addresses and Port 1 reads Address1 and Port 2 reads Address 2. When the ports want to write to different addresses (Address1 ≠ Address2) the data is being written both the ports to their addresses. Finally in the event they both write into the same address (Address1 = Address2) then Port 1 will write into Address1 and Port 2 will write no data. This is a flexible mechanism that makes sure that the system can serve a variety of read/write cases, and hence data integrity is preserved, and the access to the memory is optimized.

#### IV. RESULTS & DISCUSSION

##### A. Results

Functional simulation plays a very important role in checking the functionality of the design and checking whether it is fulfilling all the system specifications. The system was tried with five different test cases and with each of them, one has the corresponding waveform to see the behavior of the system during various operations. First of all, the memory is supposed to be empty.

➤ **Test Case 1: Port 1 Sends Data; Port 2 Receives Data.**

In this version, Port 1 uses the memory as a writer, whereas Port 2 uses it as a reader. The write-read enable (wr\_1) of Port 1 and (wr\_2) of Port 2 is set to one and zero respectively.

- First Clock Cycle: Port 1 sets 11h to memory address 02h and Port two preloads data but does not do anything.
- Second Clock Cycle: Port 2 reads 11h at address 02h, whereas Port 1 writes 0fh at address 03h.
- Fig. 3: Waveform Illustration: Fig. 3 illustrates this interaction with Port 1 writing and Port 2 reading.

- *Test Case 2: Port 2 Data write, Port 1 Data read.*

Here, Port 2 is used to place data in memory, and Port 1 is used to get it out of memory. Port 1 write-read enable (`wr_1`) is set to 0 and the Port 2 write-read enable (`wr_2`) is set to 1.

- First Clock cycle: Port 2 writes 11h to address 22h and Port 1 is idle.
- Second Clock Cycle: Port 1 loads 11h at address 22h, and Port 2 loads 0fh at address 33h.
- Waveform Fig: Fig. 4 represents the writing and reading sequence between two ports.

➤ *Test Case 3: Read on Two Ports at the Same Time.*

In this test, the read operation is performed simultaneously on both Port 1 and Port 2 and the write-read enable signals (wr\_1 and wr\_2) are also set to the value of 0.

- First Clock Cycle: Port 2 transfers 11h to address 22h, but Port 1 does nothing.
- 2nd Clock Cycle: Port 1 and Port 2 simultaneously read 11h at address 22h.

Waveform Illustration Waveform Waveform The simultaneous read operation is illustrated in Fig. 5, where both ports are in the read state simultaneously.

➤ *Test Case 4:*

The next case of the test is simultaneous write to different addresses.

Here, the write-read enable signals (wr 1 and wr 2 ) are set to 1 and both Port 1 and Port 2 are in a simultaneous write-read operation to different memory addresses.

- First Clock Cycle: Port 1 writes 32h to address 15h, and Port 2 writes 45h to address 22h.
- Second Clock Cycle: Both ports cease to write ( $wr_1 = 0$ ,  $wr_2 = 0$ ) and now read the opposite addresses. Port 1 gets 45h of address 22h and Port 2 gets 32h of address 15h.
- Waveform Illustration: This figure of speech is presented in Fig. 6 where both ports initially do write operations and then read.

- *Test Case 5: Concurrent Write to the Same Address.*

This last test case shows what will occur at the instance where Port 1 and Port 2 both seek to write data to the same memory address. The write-read enable signals (wr\_1 and wr\_2) are positioned at one.

- First Clock Cycle: Port 1 loads 01h to address 33h, Port 2 loads 22h to address 33h.
- Second Clock Cycle: Both ports terminate writing ( $wr_1 = 0$ ,  $wr_2 = 0$ ) and seek to read address 33h. To avoid the write conflict, however, only Port 1 (01h) data is read with the Port 2 (22h) data being ignored.
- Waveform Illustration: This is the case that demonstrates the conflict resolution and that is represented in Fig. 7 where the data of Port 2 is ignored.

- *Test Case 1: Port 1 Writes Data, Port 2 Reads Data:*

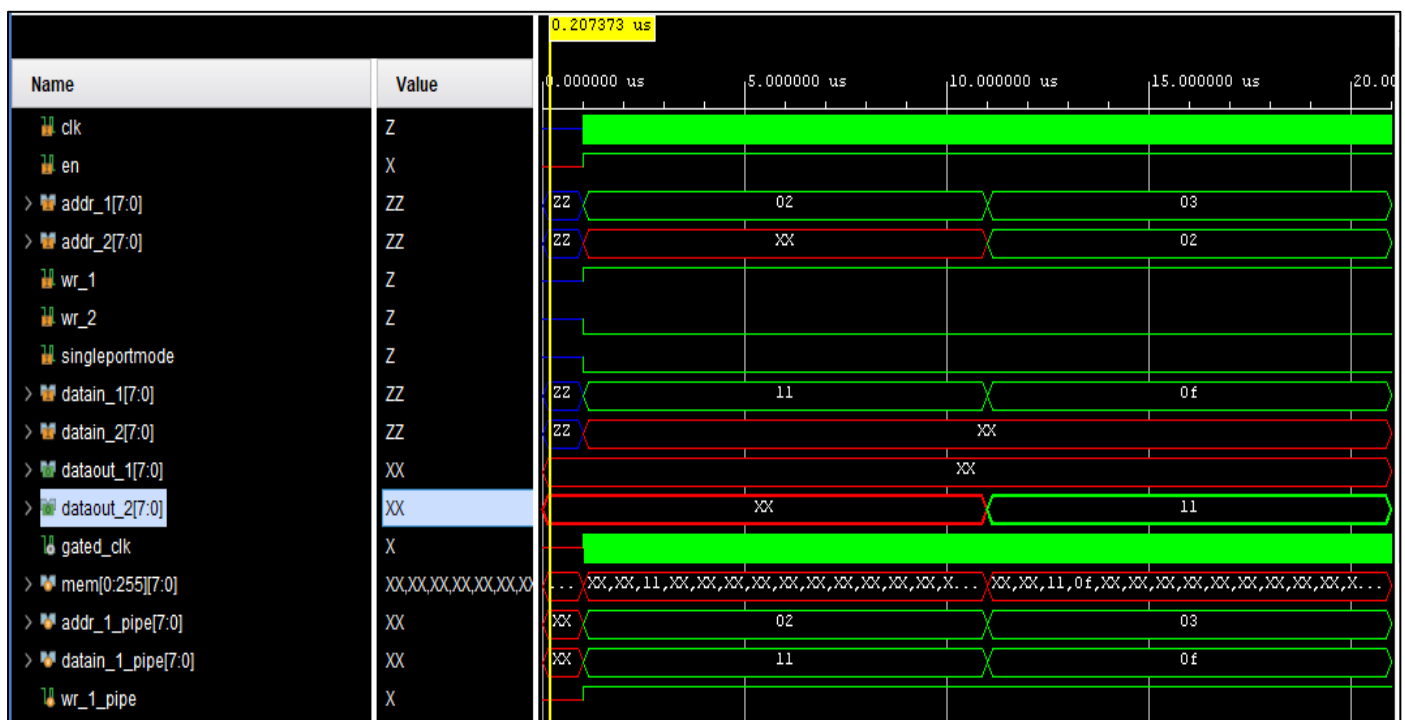


Fig 3 Writing from Port 1 and Reading from Port 2



➤ *Test Case 2: Port 2 Writes Data, Port 1 Reads Data*

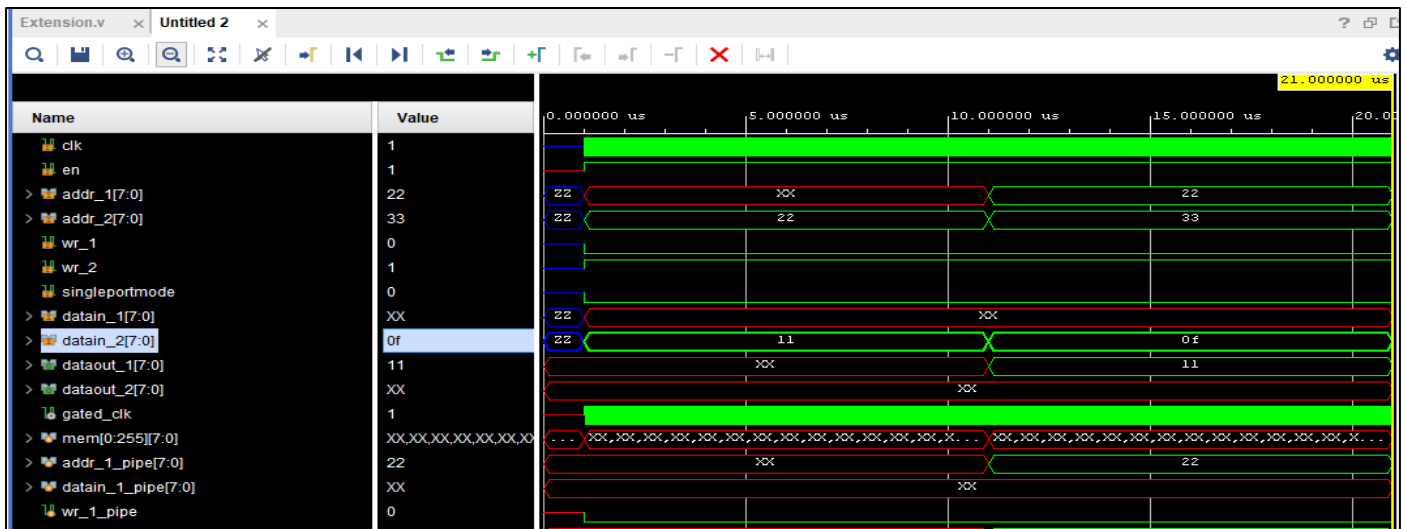


Fig 4 Reading from Port 1 and Writing from Port 2

➤ *Test Case 3: Simultaneous Read from Both Ports*

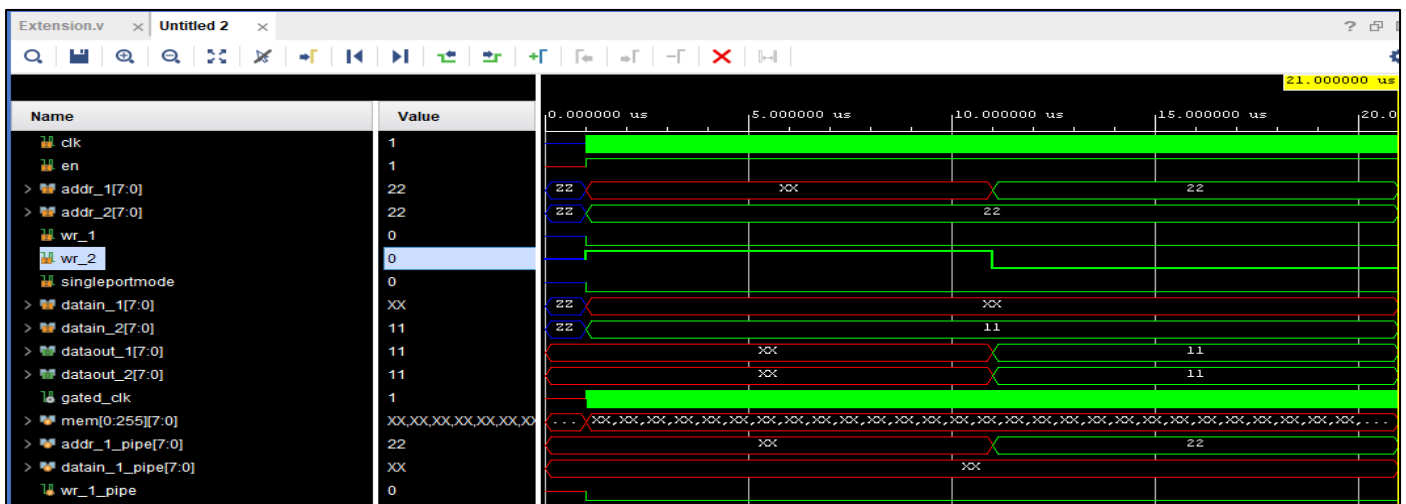


Fig 5 Simultaneous Reading from Port 1 and Port 2

➤ *Test Case 4: Simultaneous Write to Different Addresses*

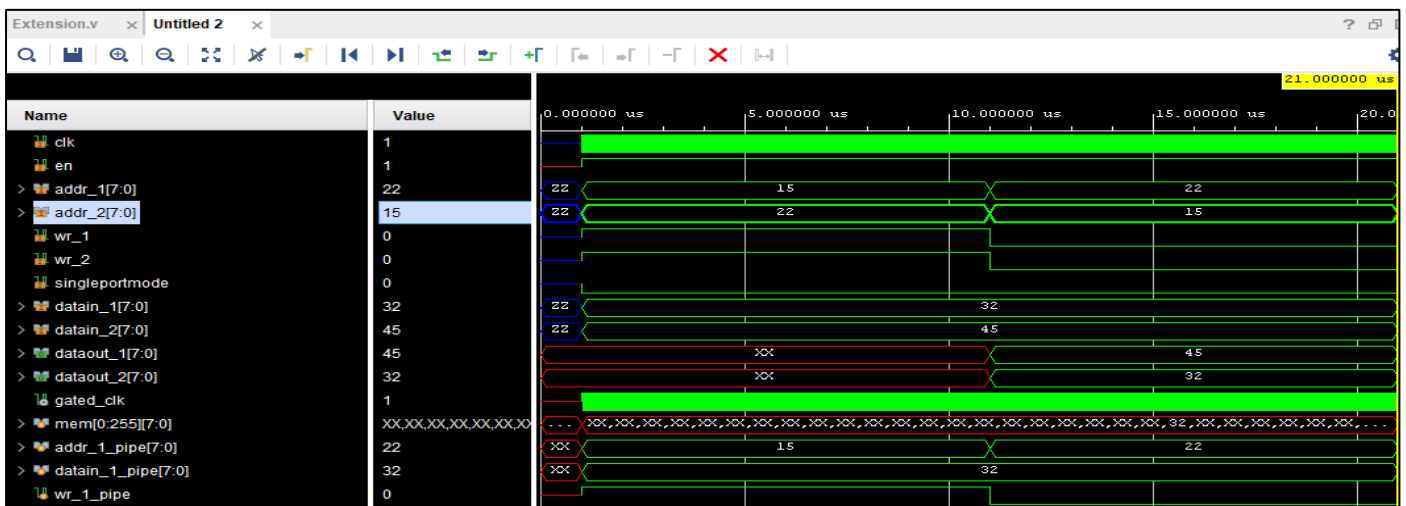


Fig 6 Simultaneous Writing from Port 1 and Port 2 to Different Addresses

➤ *Test Case 5: Simultaneous Write to the Same Address:*

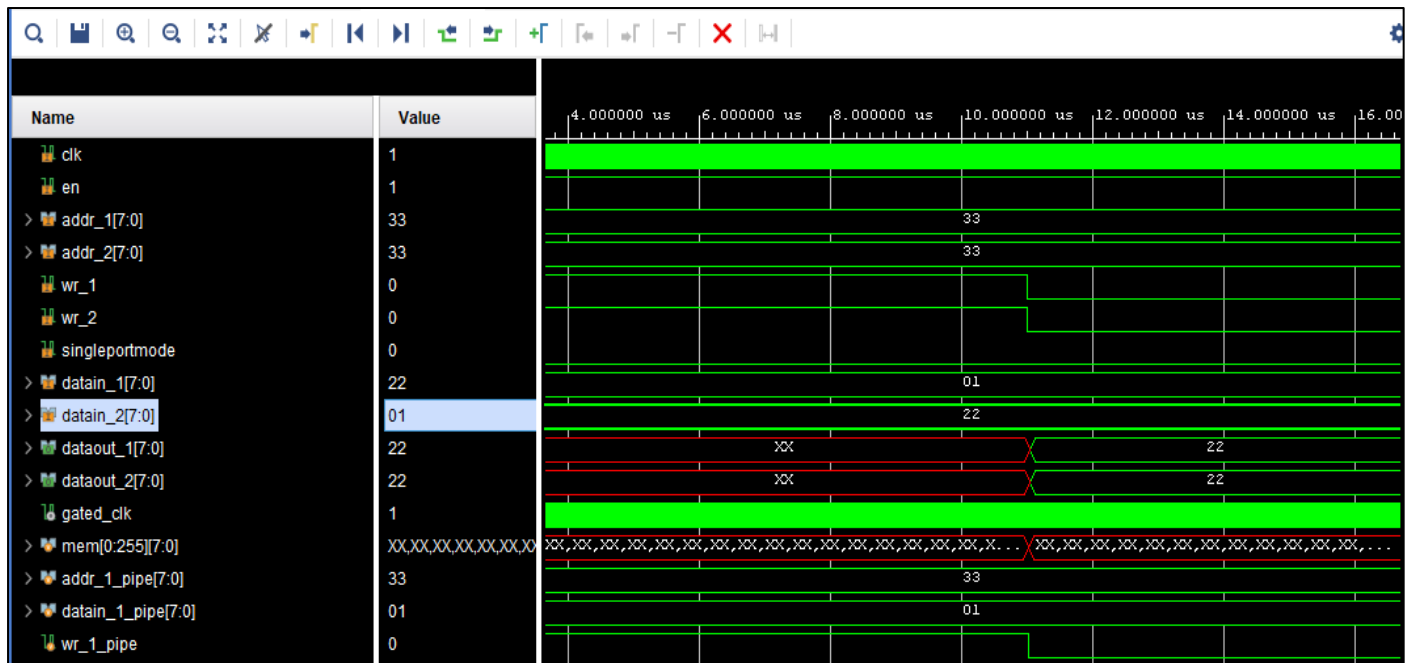


Fig 7 Simultaneous Writing from Port 1 and Port to Same Address

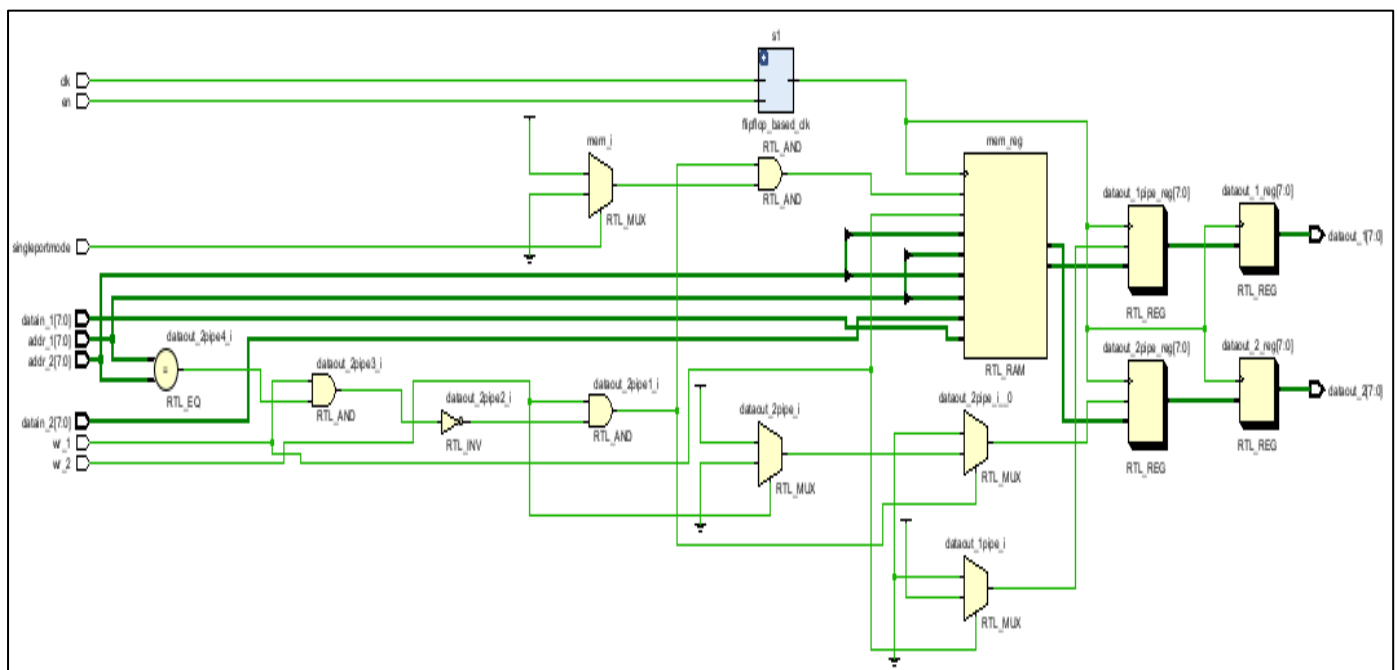


Fig 8 RTL Schematic of Efficient Dual Port Memory

This block diagram shows a control and data-path circuit where multiple input signals are decoded, processed through logic gates, and fed into a central controller. The

controller generates four output signals, each selected through multiplexers based on the control logic.

| Name                    | ^1 | Slice LUTs<br>(20800) | Slice Registers<br>(41600) | Block RAM<br>Tile (50) | Bonded IOB<br>(106) | BUFGCTRL<br>(32) |
|-------------------------|----|-----------------------|----------------------------|------------------------|---------------------|------------------|
| Extension               |    | 15                    | 35                         | 0.5                    | 53                  | 2                |
| s1 (flipflop_based_clk) |    | 0                     | 1                          | 0                      | 0                   | 0                |

Fig 9 Area of Efficient Dual Port Memory

The area report shows that the Efficient Dual-Port Memory design uses only 15 LUTs, 35 registers, and half a Block RAM tile, indicating very low resource consumption.

It also requires 53 I/O pins and 2 BUFGCTRL blocks, demonstrating an area-optimized implementation on the FPGA.

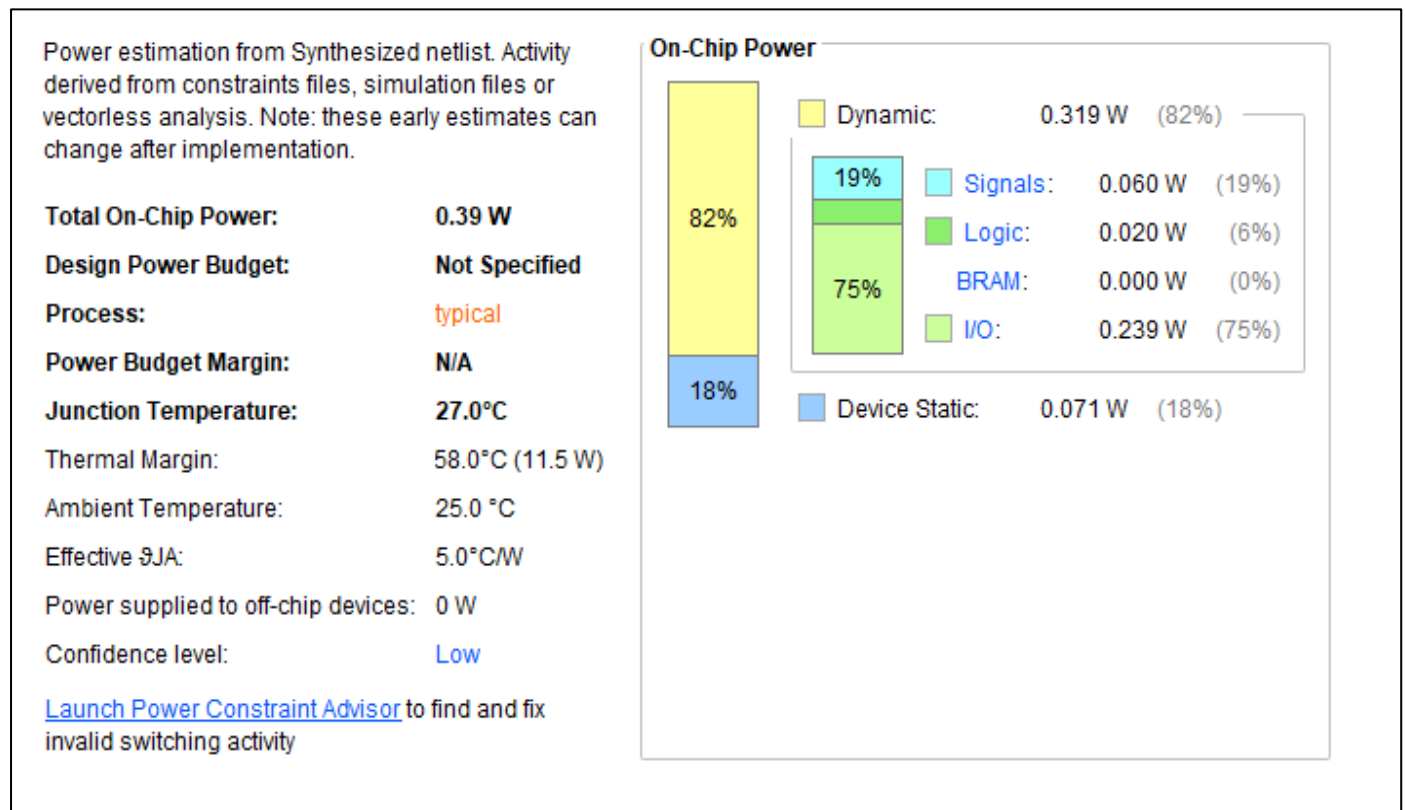


Fig 10 Power of Efficient Dual Port Memory

The power report shows the Efficient Dual-Port Memory consumes only 0.39 W total on-chip power, with 82% being dynamic power and 18% static. Most of the dynamic power comes from I/O (75%), while logic and signals contribute minimally, indicating a power-efficient design.

| Name   | Slack <sup>^1</sup> | Levels | Routes | High Fanout | From               | To           | Total Delay | Logic Delay | Net Delay |
|--------|---------------------|--------|--------|-------------|--------------------|--------------|-------------|-------------|-----------|
| Path 1 | $\infty$            | 2      | 2      | 1           | dataout_1_reg[0]/C | dataout_1[0] | 4.016       | 3.216       | 0.800     |
| Path 2 | $\infty$            | 2      | 2      | 1           | dataout_1_reg[1]/C | dataout_1[1] | 4.016       | 3.216       | 0.800     |

Fig 11 Delay of Efficient Dual Port Memory

The timing report shows a total delay of 4.016 ns for both critical paths, with 3.216 ns from logic and 0.800 ns

from routing. This indicates the Efficient Dual-Port Memory achieves low delay and meets timing comfortably.

Table 2 Evaluation Table of Dual Port Memory

| Comparison Table: Parameter | Dual Port Memory | Efficient DPM |
|-----------------------------|------------------|---------------|
| Area [LUT's]                | 14               | 15            |
| Avg Power [W]               | 16.789           | 0.39          |
| Static Power[mW]            | 0.227            | 0.071         |
| Delay[nS]                   | 6.789            | 4.016         |
| Net Delay                   | 1.599            | 0.800         |
| Logic Delay                 | 5.189            | 3.216         |
| Address Size                | 8                | 8             |
| Data Size                   | 8                | 8             |



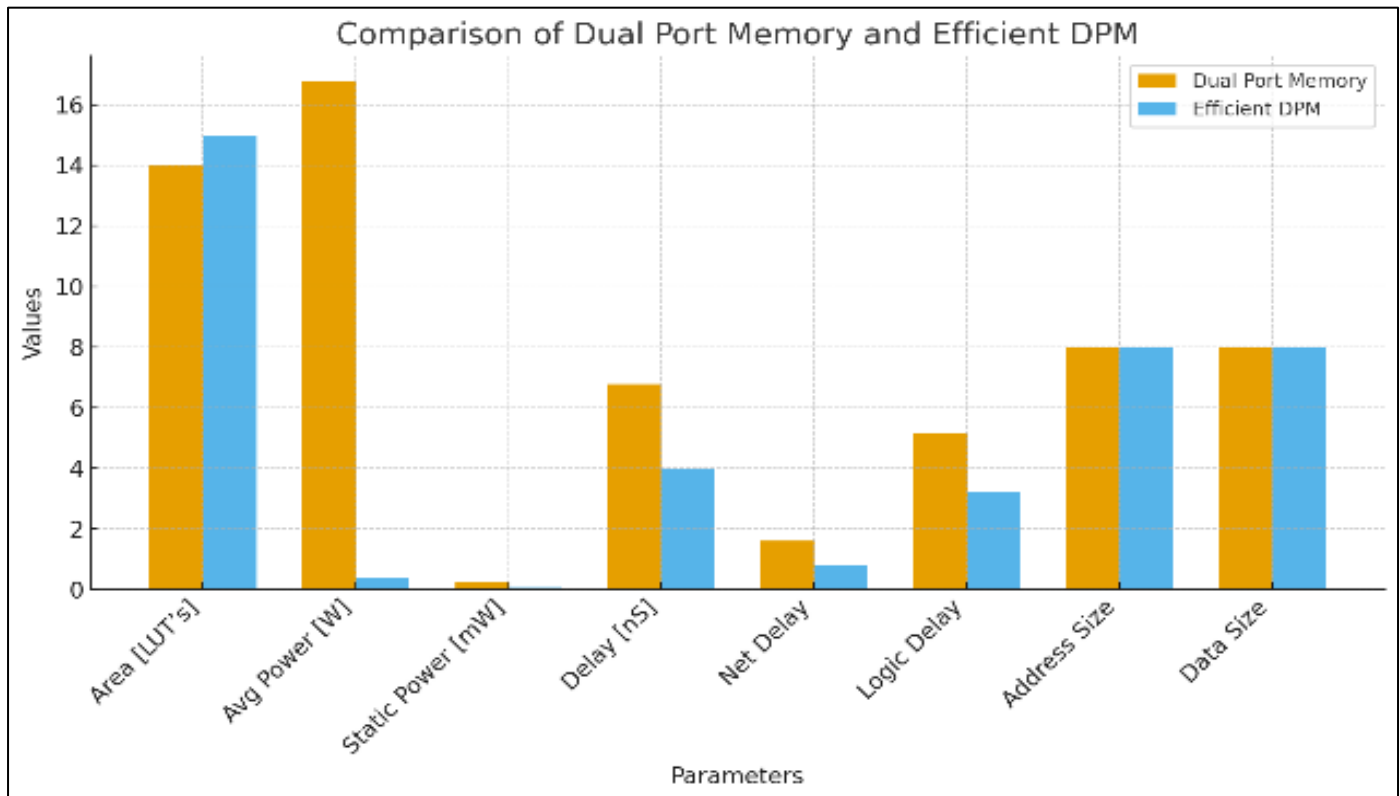


Fig 12 Comparison of Dual Port Memory and Efficient DPM

This figure presents a comparison of key parameters between the Dual Port Memory and the Efficient DPM, including Area [LUT's], Average Power [W], Static Power [mW], Delay [ns], Net Delay, Logic Delay, Address Size, and Data Size. The graph illustrates the improvements in power efficiency and timing performance achieved by the Efficient DPM.

The comparison between Dual Port Memory (DPM) and Efficient DPM demonstrates that while both systems have the same address and data size of 8 bits, the Efficient DPM offers significant improvements in both power efficiency and performance. The Efficient DPM consumes only 0.39 W of average power, a drastic reduction from the 16.789 W consumed by the Dual Port Memory, and also exhibits lower static power consumption (0.071 mW compared to 0.227 mW). In terms of performance, the Efficient DPM outperforms the traditional Dual Port Memory with a total delay of 4.016 ns, compared to 6.789 ns for the Dual Port Memory, along with reduced net delay (0.800 ns vs. 1.599 ns) and logic delay (3.216 ns vs. 5.189 ns). Despite having a slightly larger area (15 LUTs versus 14), the Efficient DPM's improved power efficiency and reduced delays make it a more optimized design overall. These enhancements position the Efficient DPM as a superior option, particularly in power-sensitive and high-performance applications.

## V. CONCLUSION

This research presents a Verilog HDL-based dual-port synchronous memory architecture optimized for both power efficiency and high performance. By employing techniques

such as dynamic clock gating, pipelined data transfer, and priority-based conflict management, the design enables concurrent read and write operations with minimal power overhead. Simulation results demonstrate substantial reductions in power consumption (from 16.789 W to 0.39 W) and improved delay performance (from 6.789 ns to 4.016 ns), while maintaining a compact area utilization. The proposed memory achieves a balanced trade-off between speed, area, and power, making it highly suitable for real-time and low-power applications in embedded systems, IoT devices, edge computing, and high-performance computing environments that demand efficient parallel data access.

## FUTURE SCOPE

Future enhancement of adaptive power scaling and higher bandwidth application support are under the scope of the current optimized dual-port synchronous memory design. Additionally, where data handling needs to be handled in real-time and at the same time, it can be integrated into more advanced platforms such as AI processors and state-of-the-art communication devices.

## REFERENCES

- [1]. M. R. Reddy, B. A. Reddy, A. Subramanyam, K. S. Reddy, D. R. Reddy, and Y. V. S. Vamsi, "Design and implementation of dual-port RAM with multi-clock support in Verilog," *Journal of Computational Analysis and Applications*, vol. 33, no. 8, pp. 3835–3842, 2024.
- [2]. A. Abumwais, A. Amirjanov, K. Uygur, and M. Eleyan, "Dual-port content addressable memory for

- cache memory applications," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 1571–1587, 2022.
- [3]. M. Jang, "Modular hardware design of pipelined circuits with hazards," *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 4, pp. 1–23, 2024.
- [4]. A. Arora, M. Kumar, and N. Sharma, "CoMeFa: Compute-in-memory blocks for FPGAs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 10, pp. 1571–1584, 2022.
- [5]. K. Marino, P. Zhang, and V. Prasanna, "ME-ViT: A single-load memory-efficient FPGA accelerator for vision transformers," *arXiv preprint arXiv:2402.09709*, 2024.
- [6]. N. S. Dhakad and S. K. Vishvakarma, "Configurable multi-port memory architecture for high-speed data communication," *arXiv preprint arXiv:2407.20628*, 2024.
- [7]. A. Al-Zawawi, K. Erdogan, and V. Kursun, "Low-power clock gating techniques for sequential circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 5, pp. 877–881, May 2012.
- [8]. A. Morgenshtein, I. Cidon, and I. Koren, "Low-power dual-port memory architecture for high-performance SoC," *IEEE Transactions on VLSI Systems*, vol. 19, no. 3, pp. 482–492, Mar. 2011.
- [9]. H. Esmailzadeh, A. Sampson, L. Ceze, and D. Burger, "Neural acceleration for approximate memory design," *ACM SIGARCH Computer Architecture News*, vol. 40, no. 3, pp. 27–38, Jun. 2012.
- [10]. Y. Kim, W. Yang, and O. Mutlu, "Ramulator: A fast and extensible DRAM simulator," *IEEE Computer Architecture Letters*, vol. 15, no. 1, pp. 45–49, Jan.–Jun. 2016.
- [11]. Y. Pan, P. Ampadu, and D. Sylvester, "Pipelined error control for low-power and high-speed on-chip interconnects," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 6, pp. 1251–1262, Jun. 2010.
- [12]. J. Abella, P. Chaparro, X. Vera, and A. González, "Low-power and high-performance adaptive memory compression," *IEEE Transactions on Computers*, vol. 59, no. 1, pp. 77–90, Jan. 2010.