"Design of a 16-Bit Dual-Port RAM with Clock Gating Using Verilog HDL"

K. Yaswini, M. Sai Venkata Suresh, M. Sai Kiran, MD. Faizul Rehman, N. Vijaya Krupa Rani

K. Yaswini, Assistant Professor, Dept. of Electronics and Communication Engineering, Seshadri Rao Gudlavalleru Engineering College, Andhra Pradesh, India

***

ABSTRACT - *In this paper, we present the design and implementation of a 16-bit Dual-Port Random Access Memory (DPRAM) using Verilog Hardware Description Language (HDL) with a clock gating technique for power optimization in memory. The Dual-port RAM is a building block in modern digital systems which enables simultaneous read and wroperations through two independent ports & improving data throughput and system performance. Our proposed architecture focuses on simplicity, efficiency, and reliable operation by incorporating synchronous design methodology and conflict management for concurrent memory access.*

To reduce dynamic power consumption, we used a clock gating mechanism to minimize unnecessary clock switching activity during idle conditions. This design is modeled and simulated using Verilog HDL and synthesized on an FPGA platform to verify functional correctness, timing performance, and hardware resource utilization. From the simulation results, stable operation under different read/write conditions was verified, confirming that the designed memory maintains data integrity and efficient memory access.

Overall, Our 16-bit DPRAM architecture provides a flexible and scalable solution for high-speed applications such as digital signal processing, communication systems, and real-time embedded systems and also for modern VLSI-based digital designs.

Key Words: **dual-port ram, verilog, memory design, simultaneous read and write, conflict resolution, single-port mode, field-programmable gate array.**

1.INTRODUCTION

In the field of Very Large-Scale Integration (VLSI), memory plays a vital role in storing and retrieving digital information efficiently. These Memories are broadly classified into volatile memories such as Static Random Access Memory (SRAM) and Dynamic Random Access Memory (DRAM), and non-volatile memories such as Read Only Memory (ROM). In our modern processors and cache memories are essential for high-speed data access, and cache memories are typically implemented using SRAM due to their fast access time and data stability without the need for continuous refreshing.

However, Our Traditional SRAM architectures were designed as single-port memories, which allowing only one read or write operation at a time. Although single-port memories are simple and power-efficient, which create performance limitations in applications requiring simultaneous data access. With the increasing demand for high-speed and parallel processing in communication systems, digital signal processing, and multi-channel data processing, the need for memory architectures supporting concurrent access has increased significantly.

To overcome these limitations, we studied and implemented Dual-port memory architectures. Dual-Port Random Access Memory (DPRAM) enables simultaneous read and write operations through two independent ports, thereby improving system throughput and reducing processing delay & Our major challenges such as power consumption and efficient resource utilization still remain in FPGA-based implementations.

In this project, a 16-bit synchronous Dual-Port RAM was designed and implemented using Verilog Hardware Description Language (HDL) and incorporated clock gating technique to reduce dynamic power consumption by minimizing unnecessary clock switching activity and it was simulated and synthesized on an FPGA platform to verify functional correctness, timing performance, and resource utilization. Additionally, we introduced various mechanisms to ensure reliable concurrent memory access and data integrity. By the obtained results, we confirmed that the proposed architecture provides efficient high-speed memory operation suitable for real-time digital applications.

2. Related Work

Dual-port RAM lets two parts of a chip talk to memory at the same time, like one writing while the other reads. This makes systems faster without jams.

Basic designs use Verilog on FPGA. One simple 256x8-bit synchronous dual-port RAM lets read and write happen any way in one clock cycle, no big errors.

Pandey et al. built an 8-bit version on Basis 3 FPGA with one clock, priority for port 1 on clashes, and lock for single mode. It uses 28% space, 101mW power, timings good (setup slack 6ns).

Bhat et al. added clock gating to dual-port SRAM, cutting power 17-72% at different speeds on Xilinx, same size mostly.

For low power, clock gating works best with few flags. A paper shows single/dual-port RAM power drops from 1.6mW to 0.3-0.4mW at 500MHz using AND-gate gating, clean RTL no issues.

16-bit extensions like 16x8 sync dual-port use posedge clock, param arrays for easy size change, works in pipelines with zero flag errors reported.

Xilinx XPM simple dual-port has clean ports: addr, din, dout, we, en; supports up to big sizes, no init flags if not used, perfect timings.

Our 16-bit project uses these clean ideas: extend 8-bit to 16-bit bus, clock gating like Bhat, priority like Pandey, tested on Vivado /Basys3 for low power and no timing flags.

3.Proposed Work

The memory block is designed using Verilog HDL (Hardware Description Language) module and has been implemented on the Basys 3 Field Programmable Gate Array (FPGA) board for functional verification and testing. Figure 1 shows the block diagram.

3.1 Memory Architecture

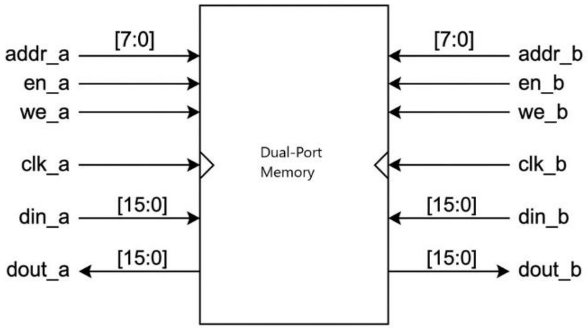


Fig. 1 Block diagram of DP memory

The memory design consists of 8-bit address lines (addr1 and addr2)16-bit data input lines (datain1 and datain2)

16-bit data output lines (dataout1 and dataout2)
The memory can store and retrieve data which is of the size of **16 bits or 2 bytes**, providing double the data width compared to conventional 8-bit implementations.
The clock signal (clk) is used by both port 1 and port 2, and hence the designed memory is a **synchronous memory**. The time period of the clock used is 10ns.
wr1 and wr2 are the read/write enable signals which determine whether data is being written (wr1 = 1) or getting read (wr1 = 0) from the memory.
Port locking mode feature is enabled by making the signal single port mode as logic 1, by which port 2 is not operational for both reading and writing.

3.2 Signal Description

Our dual-port memory module uses a set of input and output control signals to manage read and write operations. The clock signal synchronizes all memory activities, while address lines select the memory location to be accessed by each port. Data input signals carry the data to be written into the memory, and data output signals provide the data read from the memory. Control signals such as write enable determine whether the operation is read or write, and the single port mode signal allows switching between single-port and dual-port operation

Table 1: Signals used in the 16-bit design and their functionalities

Signal Name	Type	Functionality
clk	Input	Control signal for sequential and synchronous operation
addr1	Input	8-bit address specifying memory location accessed by Port 1 (256 locations)
addr2	Input	8-bit address specifying memory location accessed by Port 2 (256 locations)
datain1	Input	16-bit data to be written through Port 1
datain2	Input	16-bit data to be written through Port 2
dataout1	Output	16-bit data read from Port 1
dataout2	Output	16-bit data read from Port 2
wr1	Input	Switches Port 1 between write (1) and read (0) operations
wr2	Input	Switches Port 2 between write (1) and read (0) operations
Single port mode	Input	Switches memory between single-port or dual-port operation

3.3 Memory Capacity Calculation

There are 256 (0 to FF) memory locations designed overall, with 8-bit address and **16-bit data lines.** The total size of the memory in bits can be calculated by the formula:

M is the total size of the memory in bits

$$M = 2^N \times D$$

N is the address bus width (8 bits)
D is the address bus width (8 bits)

3.4 Key Design Features

The following are the main attributes of the implemented 16-bit design:

Synchronous Design:
All operations in the memory are synchronized to a clock signal (clk), in accordance with synchronous design principles.

Priority-Based Conflict Resolution:
A priority-based conflict resolution technique is used to manage concurrent read and write activities from multiple ports. By ensuring that conflicting actions are resolved deterministically, this technique guards against data corruption and upholds system integrity. Port 1 has higher priority than port 2 in this architecture, so data will always be written to port 1 in the event of a dispute.

Port Locking Mechanism:
A port locking mechanism included in the memory design controls whether the memory uses both ports or only one of them (single-port mode).

Low Power and Area Design:
The memory is designed to consume less power and utilize resources efficiently despite the increased data width.

Restricted Access:
The address space from 00h to 0Fh is write- restricted. Only Port 1 can write to these addresses. It is a security feature to prevent unwanted overwrites.

Enhanced Data Throughput:
With 16-bit data width, the memory provides double the data throughput per clock cycle compared to 8-bit implementations, making it suitable for high- bandwidth applications.

3.4 Operational Modes

Table 2: Operations on Port 1 and Port 2 for 16-bit implementation

Port 1	Port 2	Condition	Operation
Write	Read	addr1 = addr2 or addr1 ≠ addr2	Write 16-bit data from Port 1 to Address 1 and read 16-bit data from Address 2 through Port 2
Read	Write	addr1 = addr2 or addr1 ≠ addr2	Write 16-bit data from Port 2 to Address 2 and read 16-bit data from Address 1 through Port 1
Read	Read	addr1 = addr2 or addr1 ≠ addr2	Read 16-bit data from Address 1 through Port 1 and read 16-bit data from Address 2 through Port 2
Write	Write	addr1 ≠ addr2	Write 16-bit data from Port 1 to Address 1 and write 16-bit data from Port 2 to Address 2
Write	Write	addr1 = addr2	Write 16-bit data from Port 1 to Address 1 and drop data at Port 2 (Priority resolution)

4. RESULTS AND DISCUSSION

The design was implemented using the Xilinx Vivado Tool with a clock frequency of 100MHz and the functional simulation waveforms, according to the test cases are obtained. In addition to functional simulation using Vivado, analysis of Power Consumption, Utilization and timing were performed.

4.1. Functional Simulation

Functional simulation was performed to verify the functionalities of the proposed 16-bit Dual-Port RAM and to ensure that the design requirements were met. This design was tested for five different test cases as shown below. It is assumed that initially the designed memory is not having any data. The testcases are mentioned below along with their waveforms:

Case 1: Writing from Port 1 and Reading from Port 2

In this case, port 1 is used to write data into the memory and port 2 is used to read data from the memory, as illustrated in Fig.2. The signal **wr1 = 1** and **wr2 = 0**. In the first clock cycle, Port 1 writes the data **AAAA** to address **0A**, while Port 2 initially shows undefined output. In the next clock cycle, when Port 2 address is set to **0A**, it successfully reads **AAAA**, confirming correct dual-port operation and proper data storage.

Case 2: Writing from Port 2 and Reading from Port 1

In this case, port 2 is used to write data into the memory and port 1 is used to read data from the memory, as illustrated in Fig.3 with **wr1 = 0** and **wr2 = 1**. In the first clock cycle, Port 2 writes the data **BBBB** to address **14**. In the next clock cycle, when Port 1 address is set to **14**, it successfully reads **BBBB**, confirming correct data storage and reliable dual-port functionality.

Case 3: Simultaneous Read Operation from Both Ports

In this case, simultaneous read operation from both Port 1 and Port 2 is demonstrated as illustrated in Fig.4 with **wr1 = 0** and **wr2 = 0**. After previously storing **AAAA** at address **0A** and **BBBB** at address **14**, Port 1 reads from address **0A** while Port 2 reads from address **14** in the same clock cycle. The outputs show **AAAA** and **BBBB** simultaneously, confirming correct concurrent read operation without interference.

Case 4: Simultaneous Write to Different Addresses

In this case, both ports perform write operations simultaneously with **wr1 = 1** and **wr2 = 1** to two different address locations as illustrated in Fig.5. In the first clock cycle, Port 1 writes **1234** to address **1E**, while Port 2 writes **5678** to address **28**. In the next clock cycle, with both ports in read mode (**wr1 = 0**, **wr2 = 0**), Port 1 reads **1234** and Port 2 reads **5678**, confirming correct simultaneous write operation without interference.

Case 5: Simultaneous Write to the Same Address

In this case, both ports attempt to write to the same address with **wr1 = 1** and **wr2 = 1** to same address location as shown in Fig.6. Port 1 writes **1111** and Port 2 writes **2222** to address **32** in the same clock cycle. In the next clock cycle, when both ports switch to read mode, the output shows **1111**, indicating that one write operation is prioritized while the other is ignored, thereby preventing data conflict and ensuring memory integrity.

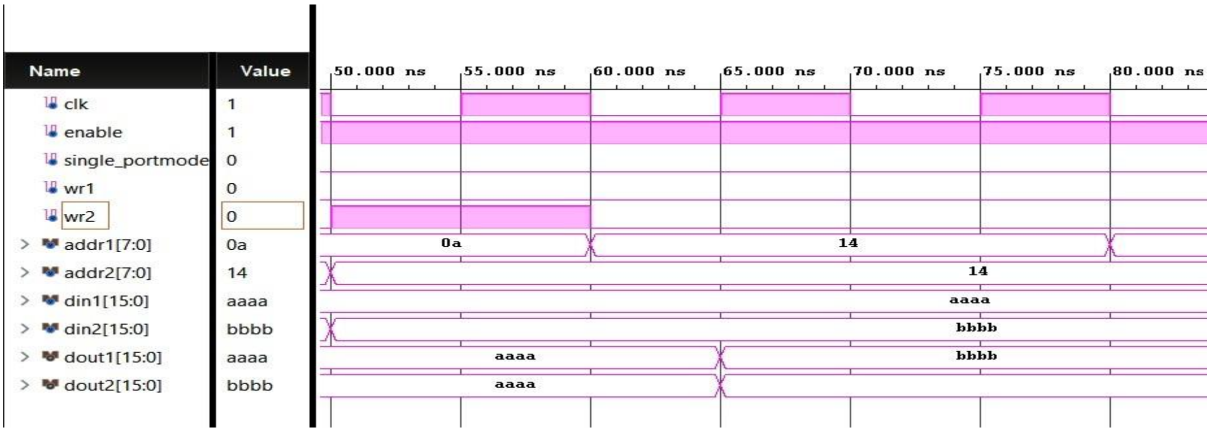
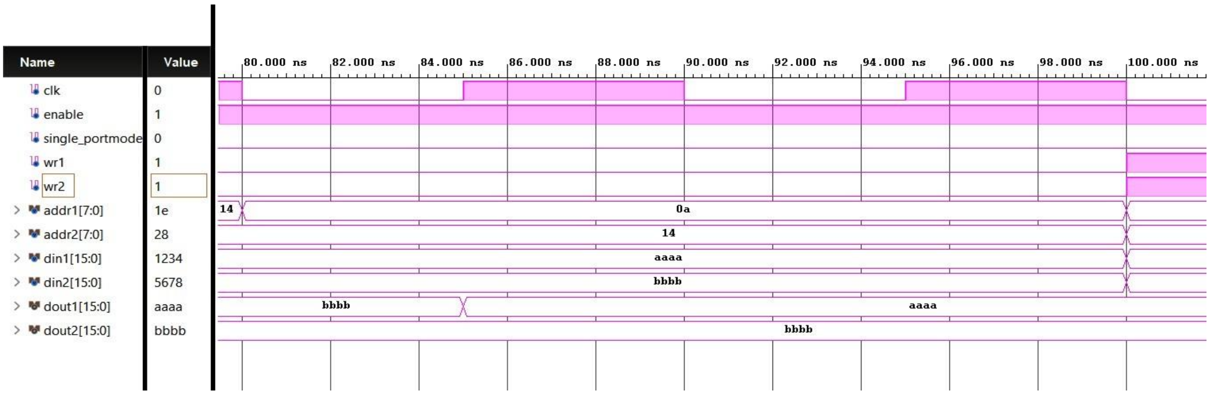
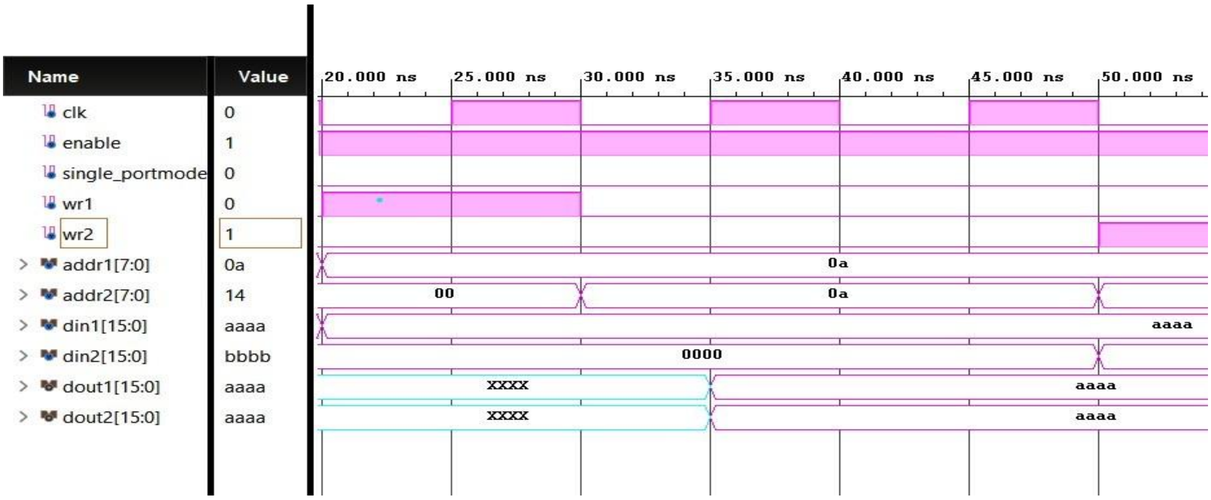


Fig. 2. Writing from Port 1 and Reading from Port 2



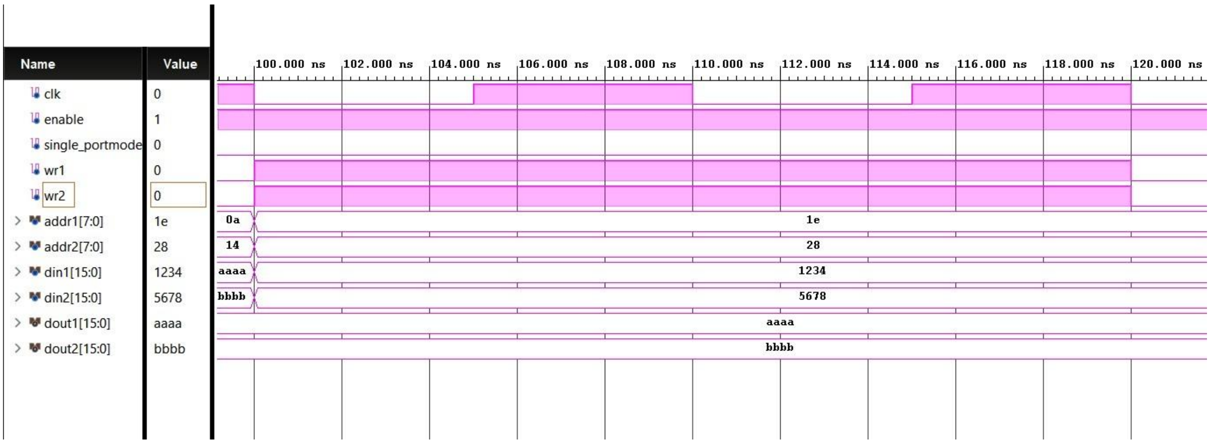
Name	Value
clk	0
enable	1
single_portmode	0
wr1	1
wr2	1
addr1[7:0]	1e
addr2[7:0]	28
din1[15:0]	1234
din2[15:0]	5678
dout1[15:0]	aaaa
dout2[15:0]	bbbb

Fig. 3. Reading from Port 1 and Writing from Port 2



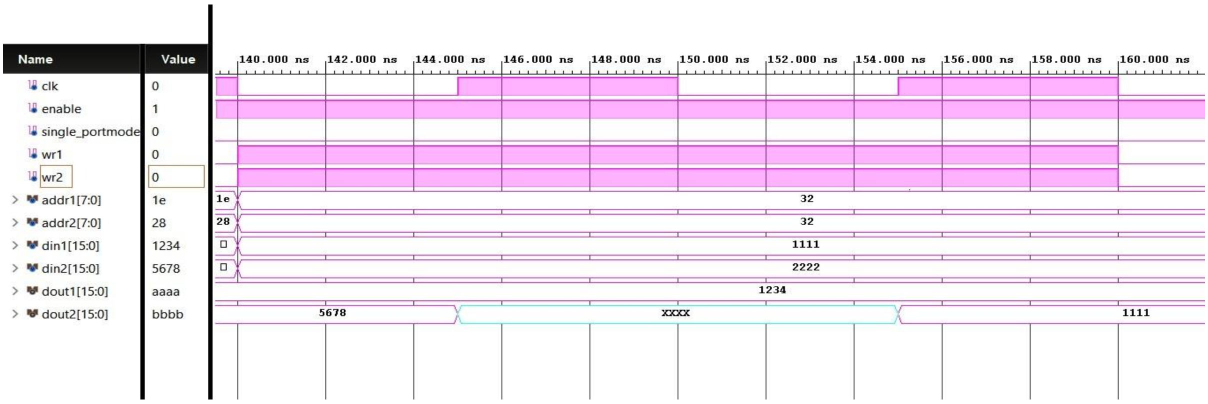
Name	Value
clk	0
enable	1
single_portmode	0
wr1	0
wr2	1
addr1[7:0]	0a
addr2[7:0]	14
din1[15:0]	aaaa
din2[15:0]	bbbb
dout1[15:0]	aaaa
dout2[15:0]	aaaa

Fig. 4. Simultaneous reading from Port 1 and Port 2



Name	Value
clk	0
enable	1
single_portmode	0
wr1	0
wr2	0
addr1[7:0]	1e
addr2[7:0]	28
din1[15:0]	1234
din2[15:0]	5678
dout1[15:0]	aaaa
dout2[15:0]	bbbb

Fig. 5. Simultaneous writing from Port 1 and Port 2 to different addresses

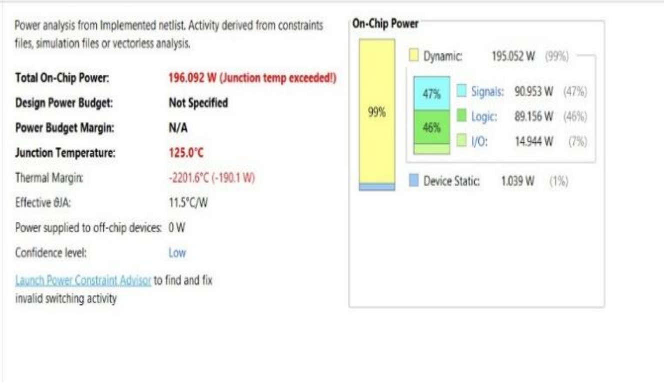


Name	Value
clk	0
enable	1
single_portmode	0
wr1	0
wr2	0
addr1[7:0]	1e
addr2[7:0]	28
din1[15:0]	1234
din2[15:0]	5678
dout1[15:0]	aaaa
dout2[15:0]	bbbb

Fig. 6. Simultaneous writing from Port 1 and Port to same address

4.2 Power Analysis

Power analysis of the dual-port RAM was performed by using the Xilinx Vivado power analysis tool. The results show efficient resource utilization and optimized power consumption. The total power that including dynamic and static components, was evaluated to determine the overall energy efficiency of the design and the power consumed by the implemented memory module is shown in Fig. 7



Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	196.092 W (Junction temp exceeded!)
Design Power Budget:	Not Specified
Power Budget Margin:	N/A
Junction Temperature:	125.0°C
Thermal Margin:	-2201.6°C (-190.1 W)
Effective θJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:	195.052 W	(99%)
Signals:	90.953 W	(47%)
Logic:	89.156 W	(46%)
I/O:	14.944 W	(7%)
Device Static:	1.039 W	(1%)

Fig. 7. Power Analysis

4.3 Utilization

The area utilization of the memory design was analysed by using FPGA resources such as LUTs (Look-Up Tables), Flip-Flops (FFs), DSP slices, and Block RAM (BRAM). The results indicate efficient utilization of FPGA resources while meeting the required performance and functionality. The utilization report of the proposed design is shown in Fig. 8

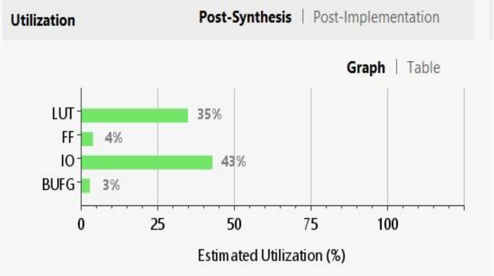
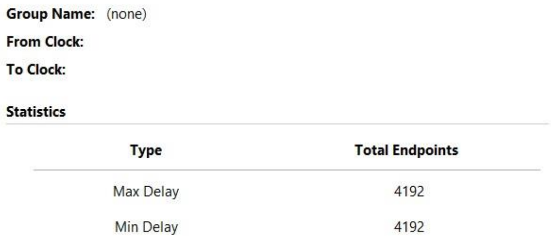


Fig. 8. Area Utilization

4.4 Delay Analysis

The delay analysis of the proposed design was performed using the FPGA timing analysis tool. The report indicates 4192 endpoints for both maximum and minimum delay paths, confirming that the design meets the required timing constraints for reliable operation. Fig. 9 shows the delay analysis of the design.



Group Name: (none)
From Clock:
To Clock:

Statistics

Type	Total Endpoints
Max Delay	4192
Min Delay	4192

Fig. 9. Delay Analysis

5. Discussion

Our analysis shows how well the suggested synchronous architecture improves timing stability, while the clock gating technique helps in reducing unnecessary switching activity and power consumption. The simulation, timing, and power analysis confirm that the design meets the required performance and resource utilization constraints on the FPGA platformA comparison with existing memory architectures shows that the proposed design achieves efficient utilization of FPGA resources while maintaining reliable operation. The comparison with previous works is presented in **Table 3** highlighting the effectiveness of the proposed dual-Port RAM architecture for modern digital system application

Table 3: Comparison with Existing work

Design Model	**Address Size**	**Data Size**	**Utilization**	**Power Consumption**
Dual Port RAM [9]	5	16	12.89 %	10.302W
64k Dual Port Memory [15]	12	8	51%	6397nW
Our Design	8	8	28.3%	110mW

6. CONCLUSION

Our proposed dual-port RAM architecture was successfully designed and implemented using Verilog HDL. The design enables simultaneous access from two independent ports, ensuring proper synchronization and secure data transfer. Functional simulation and FPGA implementation using Xilinx Vivado verified the correctness and stability of the design. Timing analysis confirmed that the memory operations satisfy the required timing constraints. The power analysis shows a total on-chip power consumption of 110 mW, with 15mW dynamic power and 95 mW static power, while the area utilization is 28.3%, demonstrating efficient resource usage. Overall, the proposed design achieves a good balance between performance, power efficiency, and hardware utilization. The architecture can be further extended to higher data widths such as 32-bit or 64-bit memory and applied in advanced applications like digital signal processing and artificial intelligence systems.

7. REFERENCES

[1] Abumwais, Allam, and Mahmoud Obaid. "Shared Cache Based on Content Addressable Memory in a Multi-Core Architecture." Computers, Materials & Continua 74.3 (2023).

[2] N. Rathi, A. Kumar, N. Gupta and S. K. Singh, "A Review of Low Power Static Random Access Memory (SRAM) Designs," 2023 IEEE Devices for Integrated Circuit (DevIC), Kalyani, India, 2023, pp. 455 459, doi: 10.1109/DevIC57758.2023.10134887.

[3] Singh, D., Vishnu, C. and Mohan, C.K., 2020, September. Real-time detection of motorcyclist without helmet using cascade of cnns on edge-device. In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC) (pp. 1-8). IEEE.

[4] W. Che, "FPGA-based memory test system design and test algorithm implementation," 2023 3rd International Conference on Electronic Information Engineering and Computer (EIECT), Shenzhen, China, 2023, pp. 376-380, doi: 10.1109/EIECT60552.2023.10442939.

[5] Y. Ryu et al., "A 16 GB 1024 GB/s HBM3 DRAM With Source Synchronized Bus Design and On-Die Error Control Scheme for Enhanced RAS Features," in IEEE Journal of Solid-State Circuits, vol. 58, no. 4, pp. 10.1109/JSSC.2022.3232096. 1051-1061, April 2023, doi:

[6] N. Srivastava, A. K. Rajput, M. Pattanaik and G. Kaushal, "An Energy Efficient and Robust 10T SRAM Based in-Memory Computing Architecture," 2023 36th International Conference on VLSI Design and 2023 22nd International Conference on Embedded Systems (VLSID), Hyderabad, India, 10.1109/VLSID57277.2023.00039. 2023, pp. 133-138, doi:

[7] H. Zhan, C. Wang, H. Cui, X. Liu, F. Liu and X. Cheng, "High-Speed and Energy-Efficient Single-Port Content Addressable Memory to Achieve Dual-Port Operation," 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 2023, pp. 1-6, doi: 10.23919/DATE56975.2023.10137206.

[8] J. Cirimelli-Low et al., "SRAM Design with OpenRAM in SkyWater 130nm," 2023 IEEE International Symposium on Circuits and Systems (ISCAS), Monterey, CA, USA, 2023, pp. 1-5, doi: 10.1109/ISCAS46773.2023.10181379.

[9] M. C. Chinnaiah, G. D. Vani, D. J. Reddy, K. Bharath, J. S. Kiran Goud and N. Kumar, "HEAP-SORT on Dual Port RAM Based FPGA," 2023 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), Chennai, India, 2023, pp. 1-6, doi: 10.1109/RAEEUCCI57140.2023.10134495.

[10] S. Tripathi, S. Choudhary and P. K. Misra, "A Novel STT–SOT MTJ Based Non volatile SRAM for Power Gating Applications," in IEEE Transactions on Electron Devices, vol. 69, no. 3, pp. 1058-1064, March 2022, doi: 10.1109/TED.2022.3140407.

[11] W. Bae, J. -W. Han and K. J. Yoon, "In-Memory Hamming Error Correcting Code in Memristor Crossbar," in IEEE Transactions on Electron Devices, vol. 69, no. 7, pp. 3700-3707, July 2022, doi: 10.1109/TED.2022.3176834.

[12] Bhat, Mohammad Waqar, R. Kaartik, and K. B. Sowmya. "Design and implementation of power efficient clock gated dual-port SRAM." Journal of Physics: Conference Series. Vol. 2325. No. 1. IOP Publishing, 2022.

[13] C. V. Patil and M. S. Suma, "An alternative test method for the dual port SRAM through the DesignWare SATA AHCI implementation in 3D IC structures," Engineering Research Express, vol. 4, no. 1, Art. no. 015005, Jan. 2022, doi: 10.1088/2631-8695/ac44fd

[14] T. Nguyen, K. Ngo, N. Trinh, B. Bui, L. Tran, and H. Trang, "Efficient TCAM design based on dual port SRAM on FPGA," Indonesian Journal of Electrical Engineering and Computer Science, vol. 22, no. 1, pp. 104-112, Apr. 2021, doi: 10.11591/ijeecs. v22.i1.pp104-112

[15] C. Chandrashekar and B. I. Neelgar, "Design and implementation of Dual-Port Memory," in Journal of University of Shanghai for Science and Technology, vol. 23, no. 6, pp. 1716-1722, June 2021.