A Fault-Tolerant Dual-Port RAM Architecture Using ECC and Conflict Arbitration

Somu Basappa Goudagavi[1*], Suhas G V[2], Amar Ghule[3], Dr.Sunitha Y N[4]

[1,2,3]Student,[4]Assistant Professor, *Corresponding Author
Department of Electronics and Communication Engineering,
SJB Institute of Technology, Bengaluru, India

Date of Submission: Dec 28, 2025; Date of Acceptance: Dec 29, 2025; Date of Publication: Jan 03, 2026

***Abstract*:** Memory reliability has become a critical concern in modern embedded and high-performance systems due to increasing data density and reduced noise margins. This paper presents the design, implementation, and FPGA validation of a dual-port memory architecture integrated with single-error correction and double-error detection (SEC–DED) capability using Hamming codes. The proposed system supports simultaneous access from two independent ports and employs a round-robin arbitration mechanism to resolve same-address write conflicts deterministically. To enhance reliability, an automatic error injection mechanism is incorporated for fault emulation, along with a scrubbing strategy that corrects detected single-bit errors and updates memory contents in real time. The complete architecture is implemented on a Xilinx Spartan-6 FPGA and validated through waveform simulation, on-board LED and LCD outputs, and functional testing using a keypad interface. Synthesis, timing, area, and power analyses are performed using Cadence Genus, demonstrating that the proposed design meets timing constraints while achieving reliable error detection and correction with acceptable hardware overhead. The results confirm the suitability of the proposed architecture for safety-critical and fault-tolerant embedded applications.

***Index Terms*:** ECC Memory, Dual-Port RAM, SEC-DED, Round-Robin Arbitration, Error Scrubbing, FPGA Implementation, Fault Tolerance

I. INTRODUCTION

Memory reliability has become an important concern in modern digital systems due to technology scaling, reduced noise margins, and increased susceptibility to transient faults. Soft errors caused by radiation effects, electrical disturbances, or process variations can corrupt stored data and lead to incorrect system behavior. Ensuring reliable memory operation is therefore essential, particularly in embedded and real-time applications where data integrity must be preserved. Dual-port memories are commonly used to improve system performance by allowing concurrent access from multiple processing units. However, supporting simultaneous access introduces challenges such as write conflicts, data consistency, and increased vulnerability to memory faults. Many existing dual-port memory implementations either do not incorporate error correction or depend on external controllers and software routines for fault handling, which increases complexity and latency.

Error correcting codes (ECC), especially single-error correction and double-error detection (SECDED) schemes, provide effective protection against transient memory faults. While SECDED techniques are widely adopted, their integration with dual-port memory architectures requires careful handling of arbitration, error recovery, and fault repair. Conventional periodic scrubbing approaches can reduce error accumulation but often introduce unnecessary memory traffic and delayed correction.

This paper proposes a fault-tolerant dual-port memory architecture that integrates SECDED, round-robin arbitration, and autonomous memory scrubbing within a unified hardware design. The proposed system

supports concurrent memory access, resolves write conflicts deterministically, and performs real-time error detection and correction without software intervention. A fault injection mechanism is included to evaluate system robustness under single-bit and double-bit error conditions.

II. Problem Statement

Dual-port memory systems face multiple challenges when deployed in fault-prone environments:

- Concurrent access to the same memory address can lead to data corruption.
- Single-bit and multi-bit memory faults may remain undetected without ECC.
- Error correction without memory update can result in repeated fault occurrences.
- Lack of visibility into arbitration and error behaviour complicates system validation.

Existing memory architectures often address these issues independently, resulting in increased complexity, higher latency, or limited fault observability. Therefore, there is a need for an integrated memory architecture that ensures data reliability, deterministic arbitration, and real-time fault recovery while remaining suitable for FPGA implementation

III. Motivation

The motivation for this work arises from the increasing demand for reliable memory subsystems in embedded platforms where hardware redundancy is constrained by cost and power limitations. Academic designs frequently demonstrate ECC concepts in isolation, while industrial systems require combined solutions that include arbitration, fault injection, correction, and monitoring. Additionally, many FPGA-based memory demonstrations lack practical fault emulation and on-board observability. This project aims to bridge that gap by providing a complete, demonstrable system where ECC behavior, arbitration decisions, and scrubbing events can be directly observed using LEDs, LCD displays, and waveform analysis.

IV. Objectives

- To design a dual-port memory architecture with integrated SEC–DED capability.
- To implement round-robin arbitration for same-address write conflicts.
- To incorporate automatic error injection for fault emulation.
- To implement a scrubbing mechanism that updates corrected memory contents.
- To validate functionality using FPGA-based visualization and simulation.

V. Hardware And Software Requirements

The proposed dual-port fault-tolerant memory architecture was implemented on a field-programmable gate array (FPGA) platform to validate concurrent memory access, error correction, arbitration, and autonomous scrubbing under real-time operating conditions. The target hardware provides sufficient logic resources and true dual-port memory blocks required to support SECDED encoding and decoding, conflict resolution, and fault recovery mechanisms.

All reliability functions, including error detection, correction, arbitration, and memory scrubbing, are implemented entirely in hardware. The architecture operates independently without the need for an external processor, firmware, or software-based memory management, ensuring deterministic timing and low-latency fault handling.

The complete system was described using synthesizable Verilog hardware description language and implemented using the Xilinx ISE Design Suite (version 14.7). The toolchain was utilized for functional simulation, synthesis, placement and routing, and static timing analysis. Post-synthesis and post-implementation reports were used to evaluate resource utilization, timing performance, and logic overhead introduced by the ECC, arbitration, and scrubbing logic.

VI. Methodology

The proposed fault-tolerant dual-port memory system is designed using a modular, hardware-centric approach that integrates error correction, arbitration, and autonomous recovery within a single architecture. The methodology focuses on ensuring data integrity under concurrent access while minimizing latency and control overhead.

A. Data Encoding and Protection

Each 32-bit input data word is protected using a single-error correction and double-error detection (SECDED) scheme. During write operations, the input data is passed through an ECC encoder that generates a 40-bit codeword by appending parity and overall check bits. This encoded data is stored directly in memory, ensuring that all retained information is protected against transient faults.

B. Dual-Port Memory Organization

The memory subsystem is organized as a true dual-port structure that supports simultaneous access from two independent ports. To allow parallel operation without unnecessary serialization, the memory is divided into even and odd banks based on address mapping. This organization enables independent read and write operations when accesses target different banks, improving throughput and reducing contention.

C. Arbitration Strategy

When both ports attempt to write to the same memory address during the same clock cycle, a round-robin arbitration mechanism is activated. The arbiter grants write access based on the previously served port, ensuring fairness and preventing starvation. Arbitration is applied only during write conflicts, allowing normal concurrent operation in non-conflicting scenarios.

D. Error Detection and Correction

During read operations, the stored 40-bit codeword is passed to an ECC decoder. The decoder recomputes parity bits, generates a syndrome, and classifies the error condition. Single-bit errors occurring in either data or parity bits are corrected in real time, while double-bit errors are reliably detected and flagged without correction. The corrected 32-bit data is then delivered to the requesting port.

E. Autonomous Scrubbing Mechanism

To prevent the accumulation of latent faults, an autonomous scrubbing mechanism is employed. When a single-bit error is detected and corrected during a read operation, the corrected codeword is automatically written back

to the same memory location. This event-driven scrubbing approach repairs memory contents immediately after correction, eliminating the need for periodic background scrubbing and reducing unnecessary memory traffic.

F. Fault Injection for Validation

A controlled fault injection mechanism is integrated into the system to validate the robustness of the error correction and scrubbing logic. Single-bit and double-bit errors are introduced into stored codewords during active read operations using pseudo-random bit selection. This allows realistic evaluation of system behavior under fault conditions without disrupting normal memory access patterns.

VII. RESULTS

This section presents the functional verification and performance evaluation of the proposed fault-tolerant dual-port memory architecture. Simulation-based analysis is used to validate correct read and write operations, arbitration behavior under access conflicts, and error detection and correction capabilities. Hardware validation is additionally performed to demonstrate real-time feasibility of the design. The results are organized to first establish baseline dual-port memory functionality, followed by arbitration performance and fault-tolerance behavior under injected error conditions.

B.Normal Dual-Port Read and Write Operation

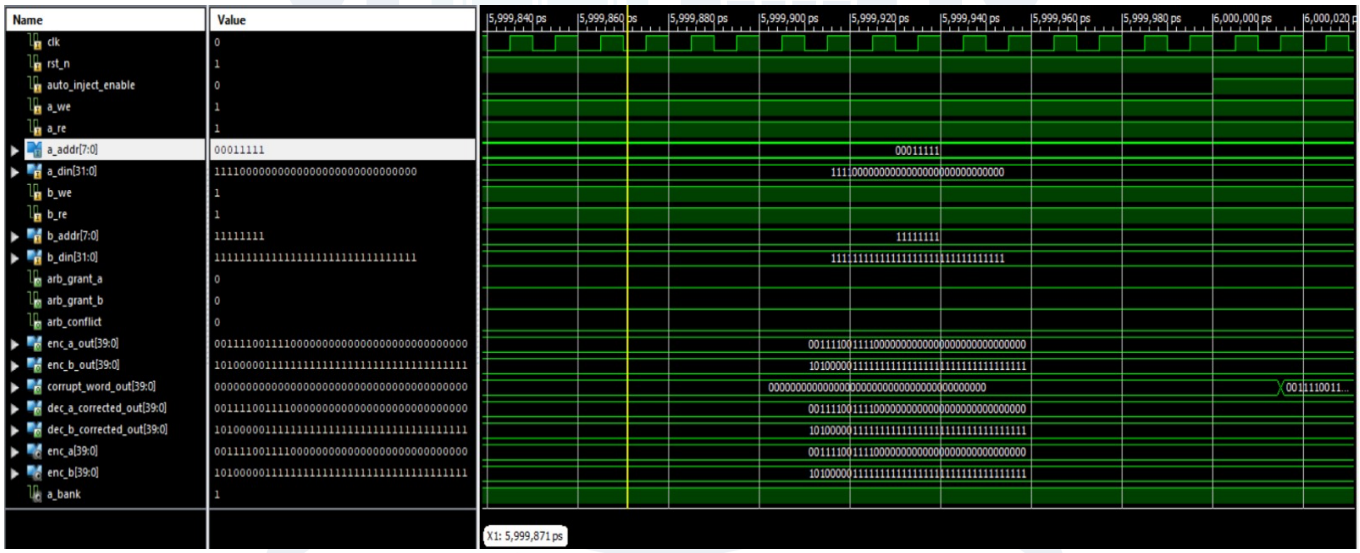

Fig. 1A. Simulation waveform of showing address and input data of both ports

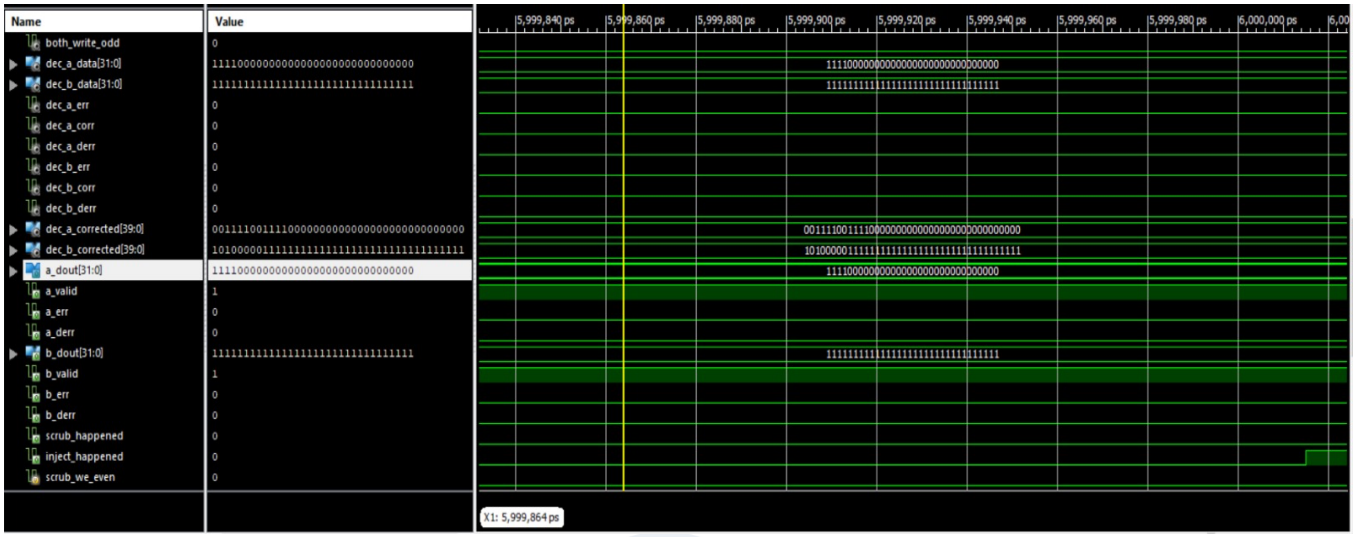

Fig. 1B. Simulation waveform of a_dout and b_dout

- Figure.1A illustrates the address signals and associated memory access behavior during fault-free operation. The waveform confirms proper address decoding and bank selection, allowing concurrent access without interference when the two ports target different memory locations. During this operation, the valid signals assert correctly, and no error flags are raised, indicating stable and correct baseline behavior.

- Figure.1B shows the simulation waveform corresponding to normal read and write operations of the dual-port memory without fault injection. Independent address and data inputs are applied simultaneously to Port A and Port B. The corresponding read outputs (a_dout and b_dout) are observed after the expected memory access latency, confirming correct data storage and retrieval for both ports.

  These results confirm the correct functionality of the dual-port memory subsystem and establish a reference point for subsequent evaluation of arbitration and error correction mechanisms.

B. Arbitration Under Simultaneous Write Access

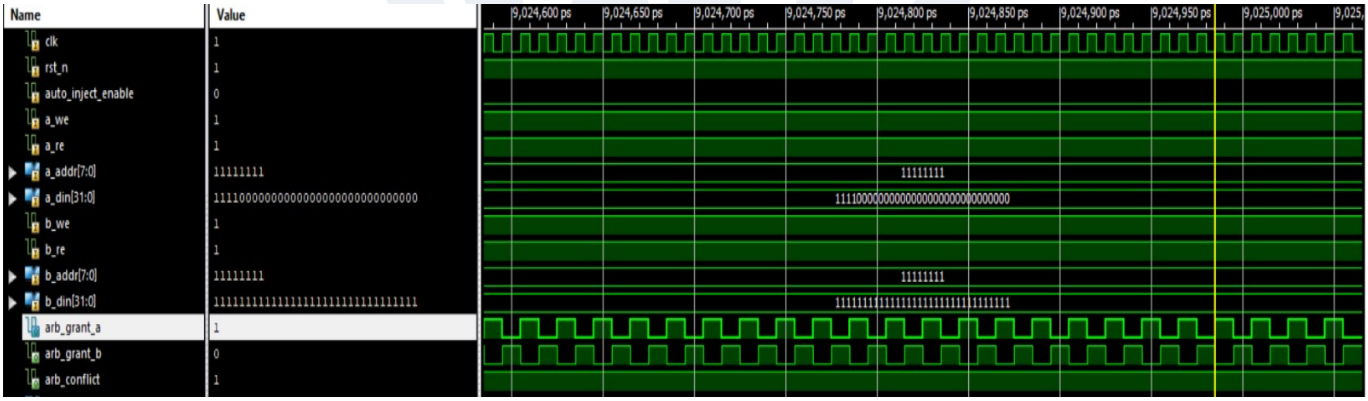

Fig. 2. Simulation waveform of Arbitration working

- Figure 2 demonstrates the arbitration behavior when both ports attempt to write to the same memory address in the same clock cycle. The round-robin arbiter resolves the conflict by granting access to one port while temporarily blocking the other, ensuring deterministic and fair access. The arbitration signals confirm correct grant generation without starvation.

C.Single-Bit Error Detection and Codeword Corruption

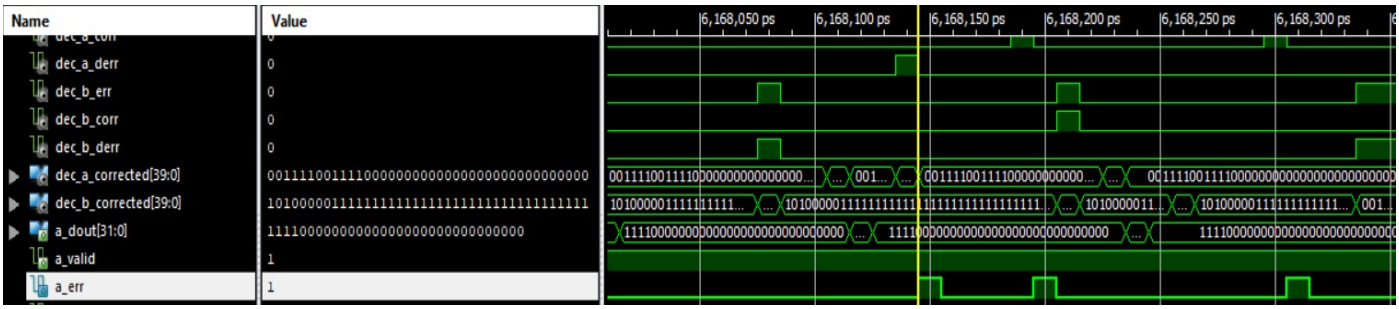

Fig. 3a. Simulation waveform showing single-bit error detection on Port A, indicated by assertion of the a_err signal during a read operation

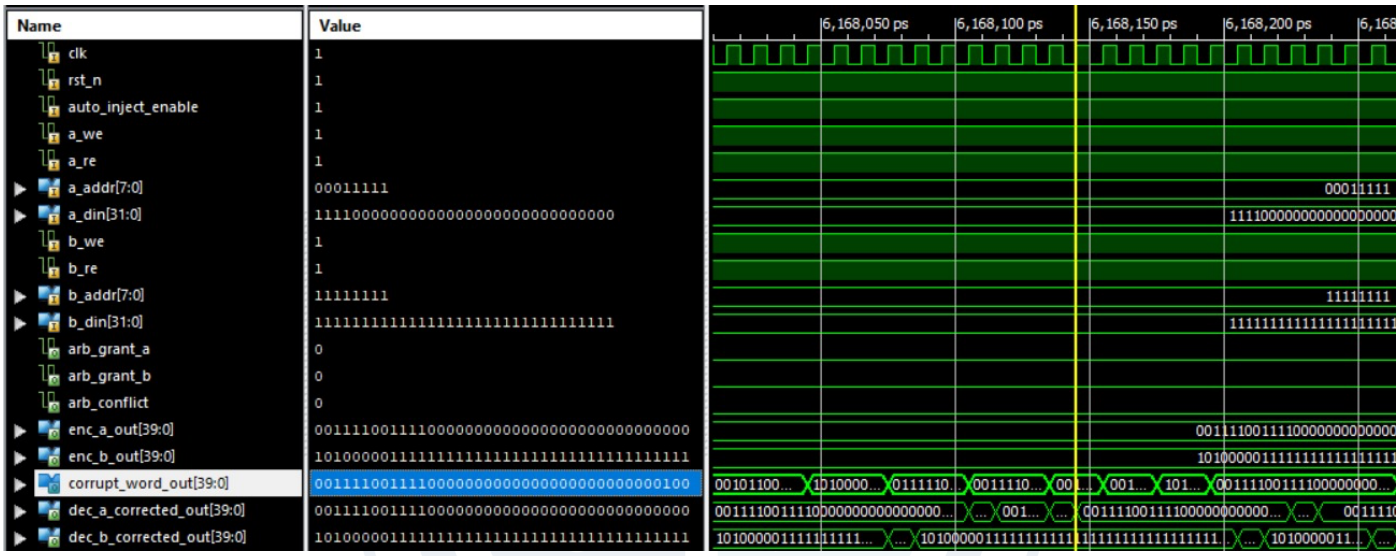

Fig. 3b. Simulation waveform showing corruption of the ECC-protected codeword corresponding to the Port A memory access following single-bit fault injection.

- Figure 3a shows the simulation results when a single-bit fault is injected into the ECC-protected memory word accessed through Port A. As observed in the waveform, the injected fault causes the Port A error flag (a_err) to assert during the read operation, confirming successful detection of an erroneous condition by the ECC decoder. The valid signal remains asserted, indicating that the read transaction completes without interruption despite the detected error.

- Figure 3b illustrates the corresponding corrupted ECC codeword generated during the same operation. The waveform clearly shows a single-bit modification in the 40-bit encoded word, confirming that the fault injection mechanism directly alters the stored ECC data. At this stage, the corrupted codeword is presented to the decoder for further processing, while no correction has yet been applied.

D. Single-Bit Error Correction and Data Recovery

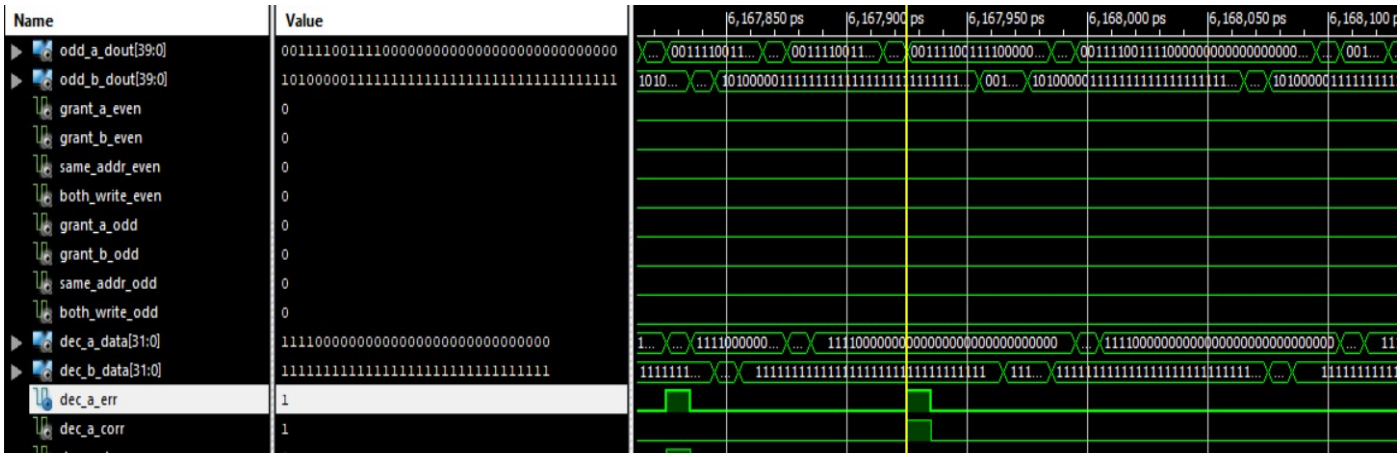

Fig. 4. Simulation waveform showing corrected and decoded data recovery after error detection

- Figure 4 presents the simulation waveform following the single-bit error detection event shown in Section C. The ECC decoder computes the syndrome corresponding to the corrupted codeword and applies the required bit correction. As observed in the waveform, the corrected ECC word matches the originally encoded value, and the recovered 32-bit data is correctly restored at the output. The correction process occurs transparently during the read operation, without requiring software intervention. The successful recovery of the original data confirms the effectiveness of the SECDED decoding logic in correcting single-bit faults within the memory subsystem.

F. FPGA Hardware Validation

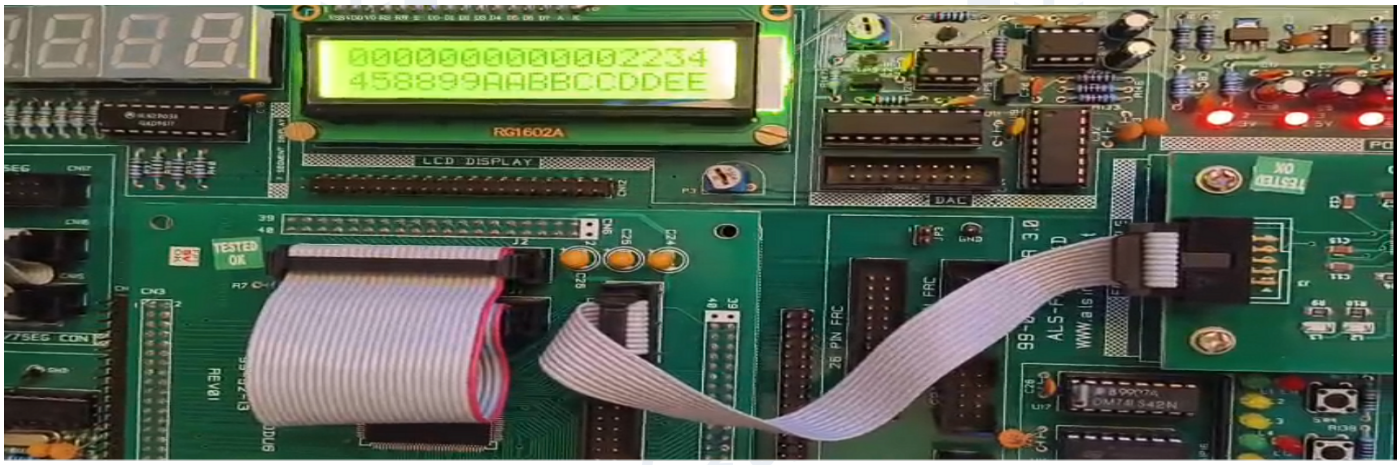

Fig. 5. FPGA hardware setup showing LCD display of combined address and data input for Port A and Port B.

- The proposed dual-port memory architecture was implemented on an FPGA platform to validate real-time functionality and interface integration. Figure 5 shows the hardware setup with the LCD interface displaying user-provided input data consisting of address and data fields for both memory ports.

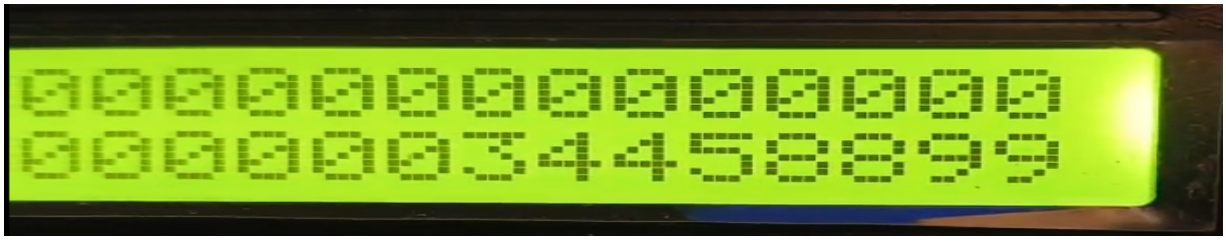

Fig. 5a. FPGA output showing correct read data from Port A during hardware operation.

Fig. 5b. FPGA output showing correct read data from Port B during hardware operation.

- Figures 5a and 5b present the read outputs of Port A and Port B, respectively, during hardware operation. The displayed values confirm correct storage and retrieval of data from independent memory ports, demonstrating proper address decoding, dual-port access, and stable operation under real-time conditions. These results verify successful synthesis, placement, and timing closure of the proposed architecture on FPGA hardware.

G. FPGA Resource Utilization

Resource Type	Used	Available	Utilization
Slice Registers	402	11440	3%
Slice LUTs	946	5720	16%
Fully Used LUT–FF Pairs	361	987	36%
Bonded IOBs	56	102	55%
Block RAM / FIFO	4	32	12%
BUFG / BUFGCTRLs	1	16	6%

Table 1. FPGA Resource Utilization

Table X summarizes the FPGA resource utilization of the proposed dual-port ECC memory system synthesized using Xilinx ISE 14.7. The design occupies a small fraction of the available logic resources, with low utilization of slice registers and lookup tables, indicating an efficient hardware implementation.

The use of block RAM resources confirms effective on-chip storage of ECC-protected memory words, while a single global clock buffer is sufficient for stable operation. Bonded I/O utilization remains within the device limits and corresponds to external interfaces such as the LCD display, keypad inputs, control switches, and status indicators. Overall, the utilization results demonstrate that the proposed architecture is well-suited for FPGA implementation with adequate margin for scalability.

VIII. CONCLUSION

This paper presented a fault-tolerant dual-port memory architecture integrating SECDED-based error correction, deterministic round-robin arbitration, and autonomous scrubbing. The proposed design supports concurrent access from two independent ports while ensuring reliable data storage and retrieval in the presence of memory faults. Functional correctness of the architecture was verified through detailed simulation results, demonstrating normal dual-port read/write operation, arbitration under access conflicts, single-bit error detection and correction, and reliable detection of double-bit errors. The autonomous scrubbing mechanism was shown to restore corrected codewords, preventing accumulation of latent faults and improving long-term memory integrity. The design was implemented on an FPGA platform to validate real-time feasibility.

Hardware results confirmed correct data entry, address decoding, and dual-port read operations, while FPGA resource utilization reports indicated low logic overhead and efficient use of on-chip memory resources. These results demonstrate that the proposed architecture is practical for implementation in embedded and safety-critical digital systems requiring reliable memory operation.

Future work may extend this architecture to support higher data widths, stronger error correction schemes, and integration with larger memory subsystems. Additionally, optimization for power and performance constraints can be explored for application-specific deployments.

IX. REFERENCES

[1] W. Wang, X. Li, L. Chen, H. Sun, and F. Zhang, *"A Review on Soft Error Correcting Techniques of Aerospace-Grade Static RAM-Based Field-Programmable Gate Arrays"*, Sensors, vol. 24, no. 16, Aug. 2024. doi:10.3390/s24165356 — comprehensive recent survey on FPGA soft error correction and scrubbing techniques, including memory-scrubbing methods.

[2] Y. Yigit, L. Maglaras, M. A. Ferrag, N. Moradpoor, and G. Lambropoulos, *"Reliability Analysis of Fault Tolerant Memory Systems"*, arXiv:2311.12849, Oct. 2023 — analyzes scrubbing strategies with SEC-DED and periodic/deterministic approaches to memory reliability.

[3] R. Burek, *"Memory Scrubbing"*, Wikipedia (ECC_memory & Memory_scrubbing) — article explaining ECC memory with SECDED and scrubbing mechanisms and how they correct soft errors in memories.

[4] "ECC Memory", *Wikipedia*, describes single-error correction and double-error detection (SECDED) ECC used in memory systems including DDR/DRAM which shares methodology with ECC RAM implemented in FPGA designs.

[5] International Research Journal on Advanced Engineering Hub (IRJAEH), *"Multi-Bit Error Resilient FPGA Cram with Minimum TTD"*, June 2024 — discusses FPGA memory reliability and scrubbing usage in harsh environments.

[6] T. Maheswaran et al., *"SEU Detection in FPGA Cram with Low Latency In-Memory ECC Checking"*, IJIRCCE, Apr. 2024 — recent FPGA ECC & scrubbing methodology focused on reducing detection/repair latency.

[7] *Frontiers in Computational Neuroscience*, *"Soft error mitigation and recovery of SRAM-based FPGAs using hybrid-grained scrubbing mechanism"*, Sept. 2023 — a verified research paper on scrubbing mechanisms and fault recovery strategies in FPGA context.

[8] Sayan Tripathi, Jhilam Jana, and Jaydeb Bhaumik, *"Implementation of Fast and Power Efficient SEC-DAEC and SEC-DAEC-TAEC Codecs on FPGA"*, arXiv:2307.16195, Jul. 2023