Design of a Memory Integrity Test System for the Safety Grade PLC(POSAFE-Q)

Ka-Ram Park [a*], Minhyuk Yoon [a], Kwan-Woo Yoo [a], Dong-Yeon Lee [a]
[a] [a]*SOOSAN ENS Co., Research Institute, 2F, Soosan Building, 13, Bamgogae-ro 5-gil, Gangnam-gu, Seoul, 06367*
[*]*Corresponding author: psh6514@soosan.co.kr*

***Keywords :** Safety Grade PLC, POSAFE-Q, Memory Integrity

1. Introduction

Safety-grade PLC in nuclear power plants monitor and control critical systems to ensure stable operation[1]. Any malfunction in these devices can lead to severe consequences, highlighting the importance of maintaining high reliability. Memory is a key component in these devices, responsible for computation and data sharing. If a memory fault occurs, critical failures such as relay malfunctions or the loss of safety functions can ensue. This study aims to design a memory integrity test system for the CPU module of POSAFE-Q, one of the safety-grade PLC used in nuclear power plants.

2. Overview of POSAFE-Q Memory Architecture

This study selected POSAFE-Q as the target system, a safety-grade control device used in domestic nuclear power plants. The CPU module in POSAFE-Q handles real-time operation system, redundant data sharing, and operational information processing[2].

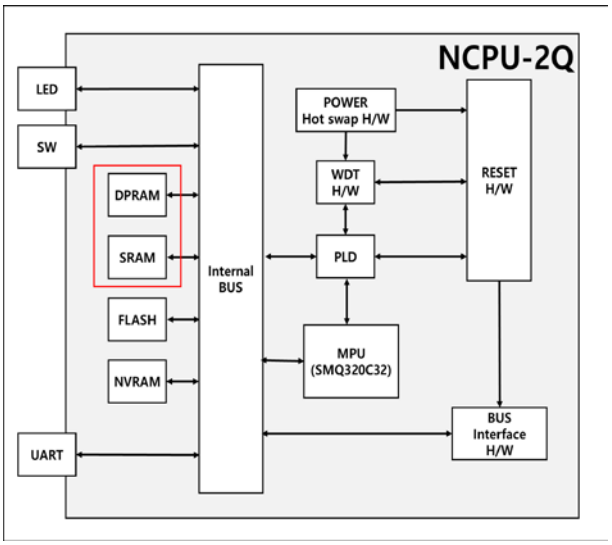

Fig. 1. Hardware Configuration Diagram of the CPU Module

Two main types of memory are crucial within this CPU module. First, SRAM is a high-speed volatile memory that stores data required for real-time operations. Next, DPRAM is a memory used for sharing data between redundant systems.

Because the POSAFE-Q module is typically interconnected with core protection logic and safety relays, even minor faults in its memory regions can escalate into full system failures. Accordingly, periodic verification of memory robustness and error management is essential, leading to a rising demand for automated inspection devices.

3. Design of the Memory Integrity Test System

3.1 System Design Overview

In this study, we designed a system that inspects the memory of the POSAFE-Q CPU module and determines whether any anomalies have occurred. The system primarily focuses on checking the status of volatile memories, such as SRAM and DPRAM[3].

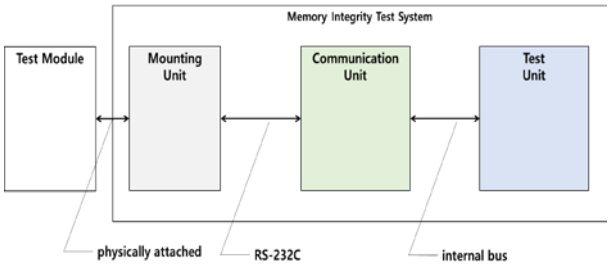

Fig. 2. Overview of the Memory Integrity Test System

The proposed system consists of a Mounting Unit, a Communication Unit, and a Test Unit.

The Mounting Unit is where the target module under inspection is physically attached to the system. The system and the module are electrically connected, exchanging control signals for memory testing.

The Communication Unit links the Mounting Unit and the Test Unit via RS-232C communication. It collects status information from the CPU module and sends test commands to the module.

The Test Unit is responsible for inspecting the mounted test module. It implements an algorithm that writes specific data patterns to the module's memory and reads them back.

By comparing the retrieved data with the expected values, it determines if there is any memory anomaly. If an error is detected, it analyzes the affected addresses or regions to identify the nature of the fault.

Additionally, the Test Unit stores error logs in file format, enabling easy review of fault occurrences and their frequency even after the inspection process. As a result, maintenance personnel can understand not only the normal/faulty status but also identify which address ranges are most prone to error.

3.2 Memory Diagnostic Approach

It should be noted that the proposed diagnostic process is conducted using a dedicated test firmware rather than the normal operating system. During normal operation, SRAM and DPRAM serve different functional roles; however, this system focuses on evaluating the physical hardware integrity of the memory itself in an off-line state. Therefore, the system systematically applies an identical write-and-read compare method across the entire memory capacities of both SRAM and DPRAM to identify physical defect patterns.

This system evaluates the memory condition through the following steps.

First is the data-writing phase. In this phase, test data is written to the target memory. The user may opt to write to the entire address range or only a specified portion.

Next comes the comparison phase. During this phase, the system reads data from the memory area previously written and checks whether it matches the test data. If a mismatch occurs, the system logs the address and data associated with the error.

After completing the comparison across all selected addresses, the system proceeds to the evaluation step. If even a single mismatched address is detected, the memory is deemed abnormal. The system then relays the error information to the user, and the diagnostic process terminates.

In particular, this system utilizes multiple data patterns sequentially, allowing it to detect a wide range of fault types rather than focusing on a single specific defect. Moreover, enabling a repetitive test feature increases the likelihood of catching intermittent faults or those sensitive to temperature/power fluctuations.

4. Results and Discussion

This section describes the practical inspection of the POSAFE-Q CPU module's memory using the proposed integrity test system. We conducted tests on both normal modules and faulty (defect-injected) modules. Additionally, we compared the outcomes with a single-pass inspection approach by utilizing the repeat inspection feature.

4.1 Result for Normal Modules

When test data were written to the memory of the normal module and subsequently read back, all addresses matched the expected values. Hence, no anomalies were reported, verifying that the fundamental inspection logic operates correctly.

Moreover, results from multiple inspection cycles similarly showed no detected anomalies in every test iteration.

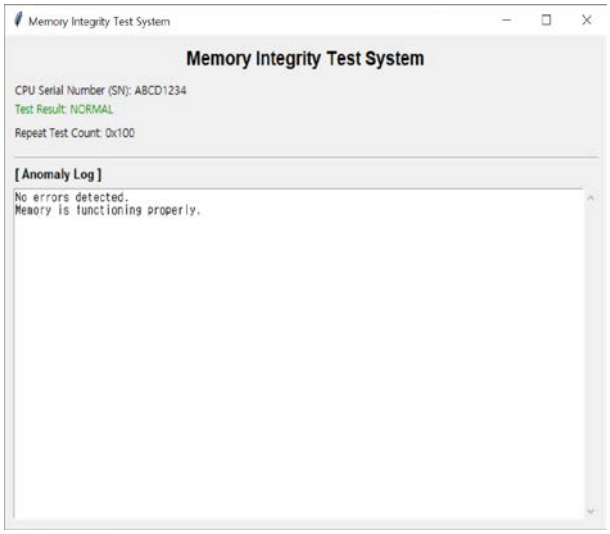

Fig. 3. Example of detected match between stored data and test data

4.2 Result for Faulty Modules

A module with artificially injected faults exhibited mismatches at multiple memory addresses. Below is a simplified illustration of the mismatched data pattern.

The proposed system automatically identified these mismatch addresses and determined that the module's memory was abnormal. This result confirms that the proposed system can effectively detect and isolate real memory faults.

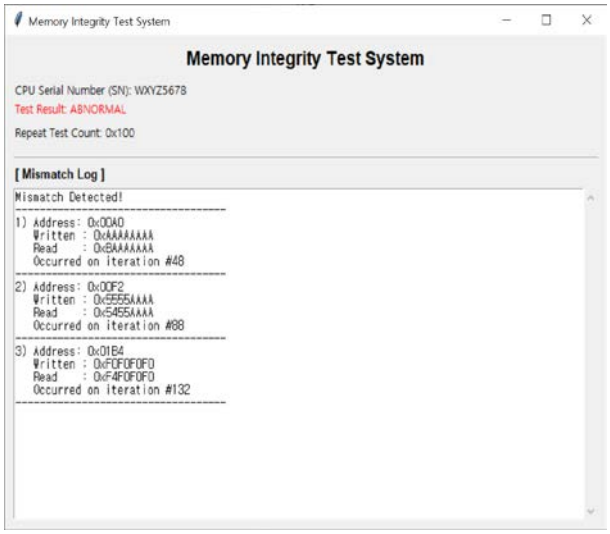

Fig. 4. Example of detected mismatch between stored data and test data

Consequently, the normal module showed zero errors, whereas the faulty module presented clear evidence of defects at specific addresses. Through multiple inspection cycles, the system also captured sporadically occurring errors with a certain probability, thus improving the reliability of fault detection.

4.3 Discussion and Future Work

While the current system focuses on offline hardware integrity verification using test firmware to isolate physical memory defect patterns, existing robust control systems like the AC160 utilize a hardware-based Mirror RAM Checker (e.g., EPLD) for real-time online validation[4]. Both the AC160's continuous Working RAM and Mirror RAM comparison and our proposed method share a fundamental 'read-and-compare' diagnostic philosophy. Therefore, our future research will aim to incorporate such online diagnostic capabilities. By utilizing the memory fault patterns identified through our current offline testing, we plan to develop an advanced architecture that constantly monitors memory integrity during normal plant operation without degrading the existing safety functions, similar to the AC160's Mirror RAM checking mechanism.

5. Conclusion

In this study, we designed a memory integrity test system targeting the CPU module of POSAFE-Q, a safety-grade PLC in nuclear power plants.

Experimental results confirmed that the proposed system, utilizing predefined data patterns, can accurately differentiate between normal and faulty modules. This approach provides a practical method to detect memory failures at an early stage in nuclear control systems, thereby enhancing overall safety.

Future work will focus on integrating these diagnostic methods into an online real-time monitoring architecture.

REFERENCES

[1] KINAC/RS-015, Cybersecurity Regulation Standard for Nuclear Facilities, Korea Institute of Nuclear Nonproliferation and Control, 2016.
[2] Su-Hyun Kim et al.. “Development of monitoring system for a safety grade PLC” Proc. of KNS, Spring, 2019.
[3] SOOSAN ENS, POSAFE-Q NCPU-2Q OS Software Design Specification, POSAFE-Q-00102-D310-1.
[4] Westinghouse Electric Company, Common Qualified Platform Topical Report, WCAP-16097-NP-A Rev. 5, 2021.