Transactions of the Korean Nuclear Society Spring Meeting
Gwangju, Korea, May 30-31, 2013

Applying Hamming Code to Memory System of Safety Grade PLC (POSAFE-Q) Processor Module

Taehee Kim [a], Sung Jae Hwang [a] and Gang Min Park [a]
[a]*Research and Development Center, POSCO Nuclear Technology, Seoul, Korea*

1. Introduction

The soundness is one of the most important factors in the processor module of safety grade PLC (POSAFE-Q). Among the parts consisting process module, memory has significant influence on the soundness because they store instructions and data. If some errors such as inverted bits occur in the memory, instructions and data will be corrupted. As a result, the PLC may execute the wrong instructions or refer to the wrong data. Hamming Code[1] can be considered as the solution for mitigating this misoperation. In this paper, we apply hamming Code, then, we inspect whether hamming code is suitable for to the memory system of the processor module.

2. The processor module with hamming code

Our processor module of the PLC is constructed as shown in Fig. 1 and Fig. 2. Fig. 1 shows our actual module and Fig. 2 shows simplified block diagram. As a CPU, TMS320C6713 and 32 bit-wide memory is adopted as a CPU and main memory respectively. Furthermore, 8 bit-wide memory, that is parity memory, is added to store 6 parity bits data for main memory.

Fig. 1. The processor module with hamming code

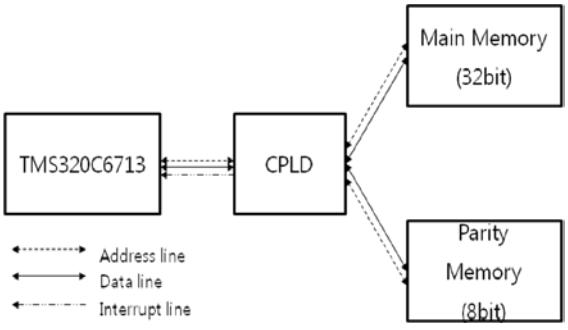

Fig. 2. The block diagram of processor module with hamming code

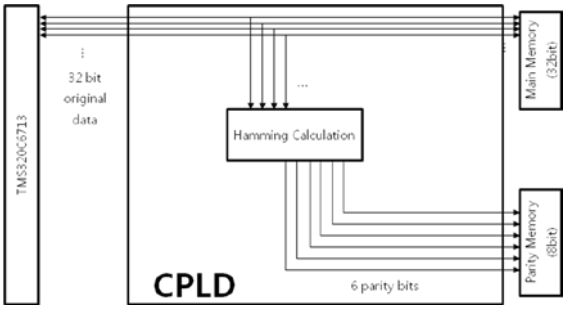

Fig. 3. Internal operation of CPLD in data writing phase (Single Error Correction)

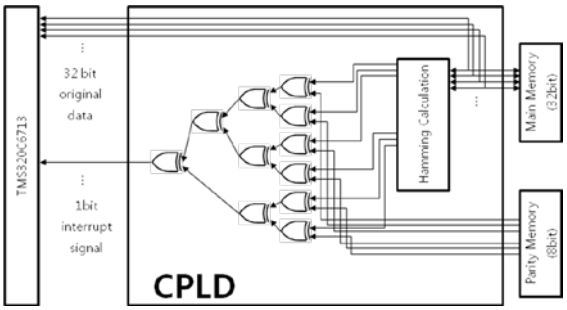

Fig. 4. Internal operation of CPLD in data reading phase (Single Error Correction)

Like shown in Fig. 2, the Complex Programmable Logic Devise (CPLD) is between the CPU and the memory. At data writing phase, it receives original data and creates parity data, then, sends data to main memory and parity data to parity memory. This phase is shown in Fig. 3. At data read phase as shown in Fig. 4, it request data to memory and execute hamming calculation after receiving data. If the calculation result is correct, CPLD regard data as sound data. Thus, it will pass data to CPU. If the result is not correct, CPLD regard data as corrupted data, so it will send interrupt signal instead of data to CPU for deriving the corrupted data process routine.

3. The implemetation of hamming code operation

It is believed that hamming code can correct 1 bit errors, detect 2 bits errors, and cannot correct or detect over 3 bits errors. Therefore, the PLC has only to correct for 1 bit errors and has only to inform CPU for 2 bits errors. For over 3 bits errors, it is beyond of hamming code capacity. Some over 3 bits errors may be detected and the other over 3 bits errors may not be detected. It varies with error bits patterns.

	parity data size	only 1bit error possible situation	under 2bit error possible situation		
		1 bit error correction and detection	1 bit error correction and detection	2 bit error detection	2bit error correction
SEC	6	○	○	X	X
SEC-DED	7 (6+1)	○	○	○	X

Table. 1. The comparative table of SEC and SEC-DED

However, above mentioned hamming code's capacity varies with version of hamming codes. For Single Error Correction version (SEC), just 1 bit error correction is possible while 2 bits errors detection is impossible. CPLD Operations in Fig. 3 and Fig. 4 describes SEC. In SEC, the syndrome, that is, the result of hamming calculation can express the existence error for the both of 1 bit and 2 bits errors. However, they cannot be distinguished between them, thus correction cannot be executed beause1 bit error correction under 2 bits error situation may causes misoperation. For Single Error Correction and Double Error Detection version (SEC-DED), 1 bit and 2 bits errors can be distinguished. However, it needs an additional parity bit compared with SEC. The example for 32 bits data is shown in Table. 1. We implemented both version of hamming code on the CPLD in the form of firmware and they will be inspected on the next chapter.

4. Inspection results

Before inspection, we have to figure out the delay time derived from memory and CPLD. Specification of memory itself is 40 ns for reading and 30 ns for writing.

Inspection results of SEC is shown in Fig. 3. Read time and write time is 70 ns and 40 ns respectively. SEC-DED shows 90 ns for read time and 60 ns for write time. Main reason for time difference between SEC and SEC-DED is the one last additional parity bit. In our 38 bit implement (32 bit original data and 6 bit parity data), CPLD has to wait total hamming calculation before the one additional parity bit for 38 bit.

Although the inspected time difference is trivial, it must be significant time delay in the PLC. All instructions and data of user application is stored in memory in the PLC. Therefore, every operations of the PLC have to refer the memory and suffer from the delay.

	Read	Write
Without hamming calculation	40 ns	30 ns
SEC	70 ns	40 ns
SEC-DED	90 ns	60 ns

Table. 2. Inspection data

5. Conclusion and discussion

In this paper, we applied hamming code to existing safety grade PLC (POSAFE-Q). Inspection data are collected and they will be referred for improving the PLC in terms of the soundness. In our future work, we will try to improve time delay caused by hamming calculation. It will include CPLD optimization and memory architecture or parts alteration.

In addition to these hamming code-based works, we will explore any methodologies such as mirroring[2] for the soundness of safety grade PLC. Hamming code-based works can correct bit errors, but they have limitation in multi bits errors.

REFERENCES

[1] R. W. Hamming, Error Detecting and Error Correcting Codes, Bell Sys. Tech. Journal, Vol.29, pp. 147-160, April 1950.
[2] D. A. Patterson, G. Gibson, and R. H. Katz, A Case for Redundant Arrays of Inexpensive Disks (RAID), University of California Berkeley, 1988.