

# 메모리 Hard Error를 극복하기 위한 메모리 Sparing 기법 설계 I : Column Sparing

구 철 회

한국항공우주연구원 통신위성연구그룹

전화 : 042-860-2145 / 핸드폰 : 017-417-7106

## Design of Memory Sparing Technique to overcome Memory Hard Error I : Column Sparing

Cheol-Hea Koo

Communication Satellite Dept. of Korea Aerospace Research Institute

E-mail : chkoo@kari.re.kr

### Abstract

This paper proposes the design technique of memory sparing to overcome memory hard error. Memory Sparing is used to increase the reliability and availability of commercial, military and space computer such as a Data Server, Communication Server, Flight Computer in airplane and On-Board Computer in spacecraft. But the documents about this technique are rare and hard to find. This paper has some useful information about memory error correction and memory error management.

### I. 서론

컴퓨터는 크게 프로세서, 메모리와 주변장치로 구분할 수 있을 것이다. 지상의 통상적인 주변 환경하에서 메모리에 에러가 발생하여 컴퓨터가 실패(Failure)할 확률은 적다고 할 수 있다. 또한 대부분의 가정용 컴퓨터나 비교적 중요하지 않은 일을 하는 컴퓨터는 메모리의 에러로 인해 컴퓨터가 실패했다고 하더라도 시간을 두고 원인을 분석하여 새로운 메모리를 구입하여 교체하면 된다. 메모리 에러의 원인으로는 정전기, ESD, EMI와 같은 전기적 요인을 먼저 손꼽을 수 있다. 그러나 금융시스템의 데이터 또는 통신 서버, 위성

탑재 컴퓨터와 같이 잠시도 컴퓨터의 기능이 중단되어서는 안되는 경우라면 메모리의 에러는 심각한 사태를 초래할 수 있는 중요한 요인이 된다.

본 논문은 메모리 에러를 검출하고 다기능적으로 정정하는 메모리 에러 관리 기법을 소개하고 있다. 특히 그중 메모리 Sparing 기법에 대한 분석, 설계, 검증 연구를 포함하고 있다.

### II. 메모리 에러 검출 및 정정

#### 2.1 메모리 에러 분류

메모리 에러는 Soft Error와 Hard Error의 2 가지로 구분할 수 있다. Soft Error는 전자기적인 원인으로 메모리 셀에 순간적인 Upset이 일어나 셀에 원래 담겨 있던 정보가 변질된 경우이다. 이 경우 주기적으로 Memory Scrubbing을 수행함으로써 에러를 검출할 수 있으며 만약 단일 bit에서 Soft Error가 발생할 경우 Memory Scrubbing 과정 중에 자동적으로 에러가 정정된다.

Hard Error는 전자기적인 원인으로 메모리 셀에 비가역적(irreversible)인 물리적, 전기적 변형이 발생되어 영구히 메모리 셀이 정상 작동을 하지 못하는 경우이다. Hard Error가 발생한 메모리 셀은 메모리 반도체 스위칭 소자의 불명확한 스위칭 동작에 의해 정보를

예측할 수 없다. 이러한 점 때문에 Hard Error는 Memory Scrubbing에 의해서 검출은 되나 정정되거나 복구될 수 없다.

## 2.2 EDAC(Error Detection and Correction)

EDAC은 ECC(Error Correction Code)와 비슷하다. ECC 기능이 램 모듈에 포함되어 있는 것과는 달리 EDAC은 메모리 컨트롤러에 포함되어 있는 것과 메모리 에러와 에러 정정에 대한 정보를 가지고 있는 잉여 메모리(Redundancy memory)를 추가로 가지고 있는 점이 차이가 있다. 본 논문에서 사용한 EDAC 알고리즘은 Hamming 알고리즘이고 데이터 16 bit에 패리티 6 bit를 첨가하여 총 코드 워드(Codeword)는 22 bit이다. 자세한 스펙은 AMD사의 Am2960 칩의 자료를 활용하면 된다. 이 Hamming 알고리즘의 메모리 에러 정정 능력은 SECEDED(Single Error Correction and Double Error Detection)이다.

메모리 에러 검출 및 정정에 대한 기능적 구성도는 그림 1과 같다.

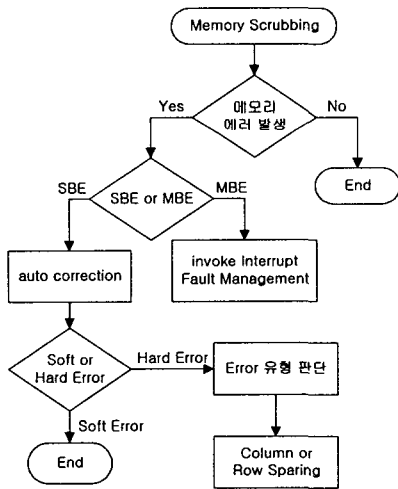


그림 1. 메모리 에러 검출 및 정정 기능적 구성도

EDAC은 Memory Scrubbing을 통하여 메모리 비트 에러를 검출하기 위해서 사용된다. EDAC을 통해서 메모리 비트의 에러 유무와 SBE(Single Bit Error)의 경우 에러가 발생한 비트 위치 정보를 알 수 있다.

MBE(Multi Bit Error)는 SECEDED 성능을 가진 EDAC으로 처리하기가 곤란하다. 이 경우 컴퓨터에 별도로 H/W 또는 S/W로 구현된 결합 허용 로직에 따

라서 처리가 되어야 한다. 물론 좀더 강력한 에러 정정 능력을 가진 BCH 코드, RS 코드, Turbo 코드 등을 사용할 수도 있으나 컴퓨터 메모리 비트 에러에 대해서는 Memory Scrubbing을 자주 함으로써 SECEDED로도 충분하다.

## III. Column Memory Sparing

### 3.1 Column Sparing

Column Sparing을 하기 위해서는 여분의 Column 비트가 필요하다. 총 요구되는 메모리의 비트의 수가 M 개라면 Column Sparing을 하기 위해서는 M+1 개의 메모리가 요구된다. Memory Sparing을 위한 전형적인 메모리 구성은 짝수 어드레스와 홀수 어드레스로 구성된 메모리 맵으로 그림 2와 같다.

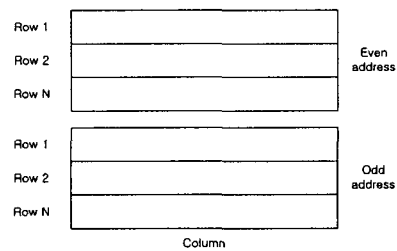


그림 2. 전형적인 메모리 구성

총 메모리는 짝수 어드레스와 홀수 어드레스로 분리되고 가로(Row)는 특정 크기로 Row 1 ~ Row N으로 나누어진다.

M+1 개의 Column 비트 중 1 개의 비트에 Hard Error가 일어나면 Column Sparing을 한다. 예를 들어 짝수 어드레스의 Column  $\alpha$  위치,  $0 \leq \alpha \leq M+1$ , 에서 Hard Error가 발생되었다면 짝수 어드레스의 모든 Column  $\alpha$  위치를 Column Sparing 비트로 치환한다. 메모리 컨트롤러의 Memory Sparing 레지스터(메모리)는 치환된 짝수 어드레스의 Column  $\alpha$  위치를 기억하고 그 후 메모리 컨트롤러는  $\alpha$  위치의 데이터를 Column Sparing 비트에서 액세스한다.

Column Sparing을 하게 되면 치환된 모든 메모리에 SBE가 발생하게 되므로 정상 기능으로 되돌리기 전에 해당 메모리에 Memory Scrubbing을 하여 에러를 정정해 주어야 한다.

Memory Sparing이 필요한 대부분의 컴퓨터가 실시간을 만족해야 하는 임무를 수행해야 하는 경우가 많으므로 실시간성을 고려하여 Memory Sparing을 하는

## 메모리 Hard Error를 극복하기 위한 메모리 Sparing 기법 설계 I : Column Sparing

데 걸리는 시간을 최소화하여야 한다.

Column Sparing의 개념도는 그림 3과 같다.

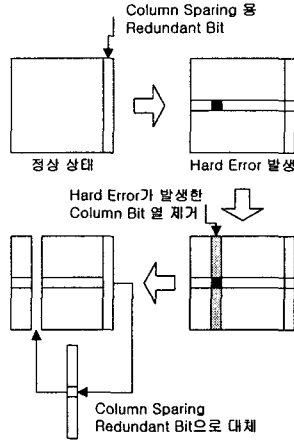


그림 3. Column Sparing 개념도

### 3.3 Memory Hard Error 검출

Memory Sparing(Column, Row Sparing)은 메모리에 비가역적 기능 상실이 발생한 Hard Error의 경우에만 적용한다. 메모리의 Hard Error를 빠르고 정확하게 검출하는 방법은 역시 EDAC을 이용하는 것이다. Memory Scrubbing을 수행하여 특정 어드레스의 특정 Column Bit에서 SBE를 검출하였다면 이것이 SBE를 처음 검출한 경우라 할지라도 Memory Hard Error에 대한 테스트를 수행하여야 한다. 왜냐하면 Soft Error 또는 Hard Error와 같은 Memory Upset이 주변 전자기 환경의 복사 피로(Radiation Fatigue)에 주된 요인이 있다고 하더라도 충분히 큰 서지 전압, 리플 전류, 역전류, 우주 복사 에너지 등에 의해서 Hard Error를 유발할 수 있는 Latch-up이 일어날 수 있기 때문이다.

Memory Scrubbing의 결과 SBE가 검출되면 에러를 정정하여 메모리에 다시 쓰기(Rewrite)를 한 후 SBE 에러의 위치를 메모리 컨트롤러 레지스터에 저장한다. 그 후 N 번째(환경 분석 또는 설계 요구사항에 따라서 변경가능)의 Memory Scrubbing에서 다시 SBE 에러가 검출되지 않았다면 이전의 SBE는 해결되었다고 볼 수 있다. 그러나 N 번째 Memory Scrubbing 이전에 다시 똑같은 위치에서 다시 SBE가 발생되었다면 이것은 Memory Hard Error로 볼 수 있으며 이 경우 상황에 따라서 Column Sparing 또는 Row Sparing을 사용할 수 있다. Memory Hard Error 검출 방식의 개념도는 그림 4와 같다.

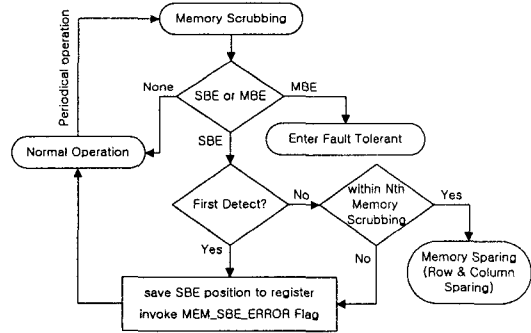


그림 4. Memory Hard Error 검출 방식 개념도

## IV. Column Sparing 검증 및 구현

### 4.1 메모리 구성 스펙

메모리는 데이터 16 비트, EDAC 패리티 6 비트, Column Sparing 1 비트, 어드레스 패리티 1 비트로 총 24 비트로 구성하였다. EDAC은 앞서 언급한 것과 같이 AMD 사의 Am2960 칩에 사용된 동일한 Hamming 코드를 사용하였다. 메모리 구성도는 그림 5와 같다.

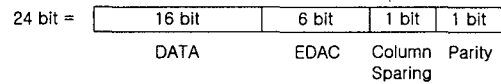


그림 5. 메모리 구성도

### 4.2 EDAC 제어기를 포함한 메모리 제어기 설계

메모리 컨트롤러는 메모리를 경유하는 모든 In/Out을 EDAC을 통하여 수행하도록 설계하였다. 메모리 제어기 구성도는 그림 6과 같다. 모든 I/O에 대해 EDAC을 통하여 데이터를 처리하는 이유는 주기적으로 수행하는 Memory Scrubbing 사이에 SBE가 일어날 수 있기 때문이며 Memory Scrubbing을 하는 이유는 SBE가 MBE로 진행하기 전에 에러를 정정하여 컴퓨터의 실패를 막기 위함이다.

Column Sparing을 검증 및 구현하기 위한 전체 구성도는 그림 7과 같다. 각 제어기는 VHDL로 코딩하였고 검증은 Xilinx Foundation Tool에서 Functional Simulation을 하여 수행하였고 그 결과 메모리 에러 정정과 Column Sparing 기능 동작을 확인하였다. 그림 8은 시뮬레이션 화면 중 일부이다.

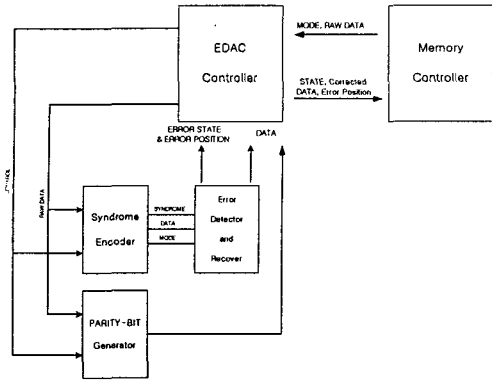


그림 6. 메모리 제어기 구성도

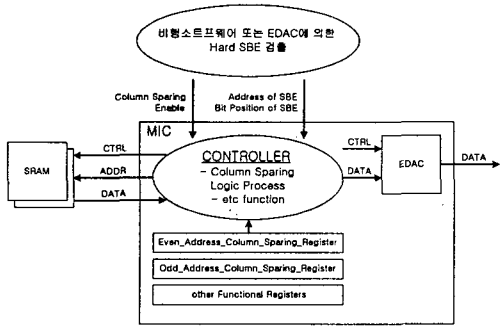


그림 7. Column Sparing 설계 구성도

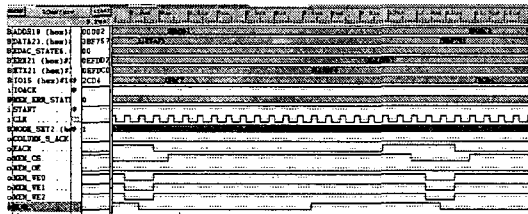


그림 8. 시뮬레이션 화면

Memory Sparing은 소프트웨어의 제어에 의해서 이루어지며 소프트웨어는 적절한 레지스터에 Memory Scrubbing 결과를 저장하고 사용자에게 리포트하고 상황에 따라 Memory Sparing을 실시한다. 본 논문에서는 위에 언급된 설계를 C/C++로 코딩하여 테스트를 완료하였다. 작은 그림 파일에 대하여 1 비트 Soft Error와 1 비트 Hard를 인위적으로 인가하여 EDAC 검증과 Column Sparing 성능을 검증하였다.

그림 9는 C/C++으로 검증한 결과이다.

### V. 결론

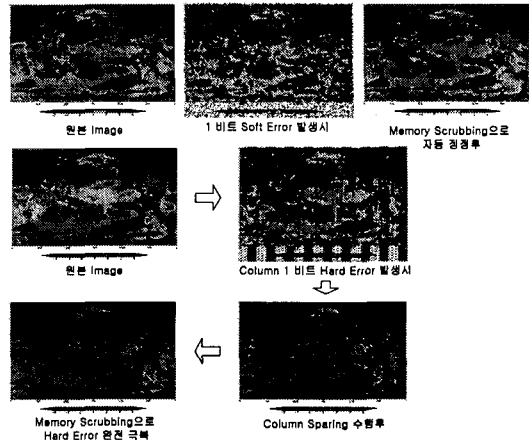


그림 9. Column Sparing 검증 결과

혹독한 주변 환경에서 운용중이거나 매우 중요한 작업을 수행하고 있는 컴퓨터는 신뢰도와 가용성을 극도로 요구하고 있다. 여러 가지 결함 허용 방안이 컴퓨터 설계시 적용되고 있으며 본 논문에서 설계한 Memory Sparing도 EDAC과 더불어 컴퓨터 메모리 부분에서 결함허용을 지원하는 기술이다. Column Sparing을 하기 위해서는 EDAC 기능이 필수적이며 1 비트 이상의 잉여 비트(Redundant Bit)가 메모리 구성시 포함되어야 한다. Column Sparing을 함으로써 메모리 에러에 의해서 컴퓨터가 곧바로 실패하는 것을 막을 수 있고 그 후 안전하게 컴퓨터를 백업 시스템을 바꾸고 Hard Error가 발생한 컴퓨터의 메모리를 교체함으로써 컴퓨터의 갑작스런 실패로 야기될 수 있는 인적·물적 손실을 최소화할 수 있다.

본 논문은 EDAC을 포함한 메모리 제어기를 VHDL로 코딩하고 시뮬레이션으로 테스트 및 검증하였고 C언어를 사용한 컴퓨터 시뮬레이션을 통해서도 그 기능성을 확인하고 Column Sparing 제어 로직을 개발하였다.

Memory Sparing은 시스템 가용성을 극도로 요구하는 컴퓨터에서 사용되며 본 논문은 고신뢰성, 고가용성 컴퓨터 시스템을 설계시 메모리 결함 허용 설계 부분에서 활용자료가 될 수 있을 것이다.

### 참고문헌(또는 Reference)

[1] ----, KOREASAT 3 Critical Design Review Data Package - Telemetry, Tracking and Command(TT&C), Lockheed Martin, 1998  
 [2] ----, KOREASAT 3 Spacecraft Operations Manual, "On-Board Computer/Flight Software, Lockheed Martin, 1999