

다목적실용위성2호용 데이터 메모리의 오류 검출 및 정정 회로 설계

조영호·심재선
한국항공우주산업(주)·삼척대학교 전기공학

Error Detection and Correction Circuit Design of Data Memory for KOMPSAT2

Cho, YoungHo Shim, JaeSun
Korea aerospace Industries LTD. Samcheok Univ.

Abstract - 다목적실용위성2호의 위성 본체시스템에는 지상과 연락을 담당하는 주 컴퓨터인 OBC, 위성의 자세를 제어 위한 원격구동장치인 RDU 그리고 위성의 전원분배를 제어장치인 ECU인 3개의 동일 프로세서(386)가 탑재되어 각 담당 임무를 수행하는 분산형 구조를 갖고 있다. 각 프로세서는 EEPROM과 SRAM 데이터 메모리를 갖고 있는데 전원 리셋이 일어나면 모든 프로그램은 EEPROM에서 SRAM으로 복사되어 운영 프로그램이 실행하도록 되어 있다. 그러나 SRAM은 우주환경에서 위성체는 방사선에 노출되어 손상을 입을 때 SEU이 발생되어 정보가 왜곡되거나 상실되는 문제를 갖고 있다. 그러므로 본 논문에서는 변형된 해밍코드 기법을 이용하여 데이터를 수신하는 곳에서 에러를 검출 및 수정하는 디지털 회로 설계방법을 기술하고자 한다.

난다. 그러므로 위성 탑재용 컴퓨터의 메모리에 대한 다른 여러 요구조건이외에도 우주 방사선에 대한 고려가 필수적으로 요구되어지고 있으며 이에 관한 연구가 많이 진행되고 있다[6-8].

다목적실용위성2호에서 SRAM에 데이터 왜곡이 발생했는지 대한 정보를 프로세서는 알아야 하고 사고에 대한 처리회로가 요구되고 있다. 그러므로 본 논문에서는 다목적실용위성2호의 프로세서보드 SRAM에서 발생하는 데이터 왜곡발생 여부를 검출하고 1비트인 경우 정정하기 위해서 변환된 해밍코드 기법을 이용하였다. 제안된 방법은 니블(nibble)단위에서 발생하는 1비트 에러인 경우에는 발생 정보와 정정할 수 있고 2비트 에러가 발생하였을 경우 발생 여부의 정보만 알려줌으로서 위성의 시스템 설계자에게 처리알고리즘을 설계할 수 있도록 하였다.

1. 서 론

우주개발 초기 단계인 1950년대 말과 1960년대에는 대부분의 위성이 이를 발사할 수 있는 발사체 능력과 위성 능력 한계로 소형위성으로 제한되었으나, 이후 발사체의 발사 능력 증가 및 위성기술의 발전과 함께 위성의 규모도 점차적으로 증가하였다. 또한 통신 산업이 발달함에 따라 상용위성에 대한 요구가 커지면서 국내에서도 독자적인 위성 개발을 위해 필요한 핵심 기술에 대한 연구의 필요성이 제기되고 있다. 특히 임무가 복잡하고 다양해짐에 따라 이를 처리해야 하는 작업이 복잡해져서 탑재 컴퓨터의 데이터 메모리의 크기는 더욱 증대되고 있다. 이러한 위성용 탑재 컴퓨터에 관한 개발 기술은 고신뢰도의 위성 개발을 위해 반드시 필요한 소요 기술로서 위성 개발 선진국들 내에서도 기술 이전을 기피하고 있는 첨단 선도 기술이다[1-5].

2. 다목적 실용위성2의 EDAC 설계

본 절에서는 데이터 왜곡의 발생을 검출하고 정정할 수 있는 기법 EDAC(Error Detection And Correction) 일부 핵심 기술을 소개하고자 한다.

2.1 EDAC 설계

제안된 방법은 데이터 1 바이트를 0, 2, 4, 6번 비트와 1, 3, 5, 7번 비트로 구성된 두개의 니블로 나누어 각 니블에 대하여 독립적으로 EDAC를 수행한 후, 결과를 조합하여 다시 1바이트로 구성하도록 하였다. 그러므로 한 워드에 대한 에러 검출 및 정정은 4비트 크기의 그룹 4개 각각에 대한 독립적인 에러 검출 및 정정이 이루어진다. 그러므로 제안한 알고리즘은 에러가 짝수 비트와 홀수 비트에서 각 각 하나씩 발생한 경우에는, 1바이트에서 발생한 2개의 에러를 검출하여 정정할 수 있게 된다.

다목적실용위성2의 위성 본체시스템에는 지상과 연락을 담당하는 주 컴퓨터인 OBC(On Board Computer), 위성의 자세를 제어 위한 원격구동장치인 RDU(Remote Driver Unit) 그리고 위성의 전원분배를 제어장치인 ECU(EPS Control Unit)인 3개의 동일 프로세서(386)가 탑재되어 각 담당 임무를 수행하는 분산형 구조를 갖고 있다. 각 프로세서는 EEPROM과 SRAM 데이터 메모리를 갖고 있는데 전원 리셋이 일어나면 모든 프로그램은 EEPROM에서 SRAM으로 복사되어 운영 프로그램이 실행하도록 되어 있다. 그러나 우주환경에서 위성체는 방사선에 노출되어 손상을 입을 때 SEU(Single Event Upset)이 발생되어 정보가 왜곡되거나 상실되는 문제를 갖고 있다. 그래서 SRAM에 1비트의 데이터 왜곡이 발생할 경우 임무수행 불능 상태가 도래될 수도 있어 위성의 수명이 끝나게 되는 최악의 상황(worst case)이 일어

그럼 1은 EDAC 기능을 수행하는 과정을 1니블만 블록도로 표시한 것이다. 프로세서가 데이터를 데이터 메모리에 쓸 경우 실제 데이터와 해밍코드에 의하여 얻어진 패리티도 패리티 메모리에 저장된다. 에러 정정 기능은 프로세서가 메모리를 읽을 경우 항상 자동적으로 수행된다. 데이터 메모리의 일정 번지를 읽으면 동시에 EDAC 메모리의 동일 번지에 있는 데이터를 읽어 들여 비교한 후, 그 결과에 따라 단일 비트 에러이면 데이터 바이트를 정정하여 프로세서에 제공하고 다중 비트 에러이면 읽어 들인 데이터를 그대로 프로세서로 제공한다. 1비트 에러는 EDAC의 기본 단위인 1니블에서 1비트의 오류가 발생한 경우에는 그림 1에서 Error Correction and

Output 블록에서 데이터를 정정하여 프로세서 공급기능 외에 수정된 데이터를 메모리에 되쓰기(writeback)라는 기능이 수행된다.

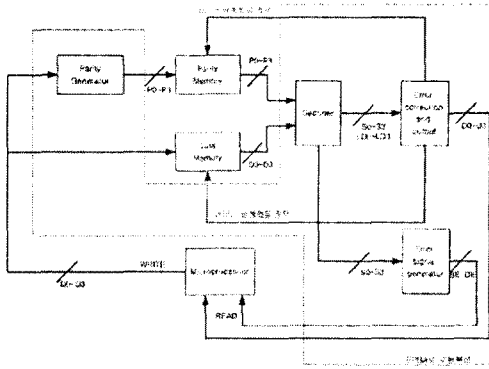


그림 1 다목적실용위성2호의 EDAC기능 흐름도
Fig. 1 Flow Diagram of EDAC for KOMPSAT2

2.2 패리티 발생부(Parity Generator)

입력 D[3:0]에 나타나는 데이터에 대해서는 무조건(타이밍 관계없이) EDAC 코드를 생성하여 P[3:0]으로 출력한다. 이런 패리티는 짝수 패리티의 해밍 코드 기법을 사용하여, 다음과 같이 데이터 4 비트로부터 3 비트를 선택하고 배타적 논리합을 취하여 4 비트의 EDAC 코드를 생성하게 하였다.

- $P3 = D0 \oplus D1 \oplus D2$; 0번, 1번, 2번 데이터 비트에 대한 짝수 패리티 비트
- $P2 = D0 \oplus D1 \oplus D3$; 0번, 1번, 3번 데이터 비트에 대한 짝수 패리티 비트
- $P1 = D0 \oplus D2 \oplus D3$; 0번, 2번, 3번 데이터 비트에 대한 짝수 패리티 비트
- $P0 = D1 \oplus D2 \oplus D3$; 1번, 2번, 3번 데이터 비트에 대한 짝수 패리티 비트

그림 2는 패리티 발생부의 일부를 실제 구현한 회로의 일부이다.

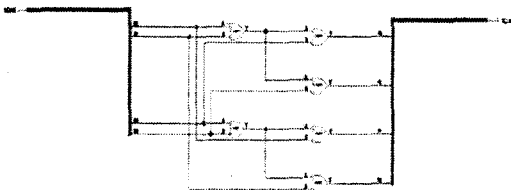


그림 2 패리티 발생부의 회로도
Fig. 2 parity generator schematic

2.3 에러정정 및 출력부

프로세서가 메모리를 읽을 경우 데이터와 패리티 코드를 읽어 들여 비교/검사하여 단일 비트 에러일 경우 정정된

데이터를 프로세서에 공급하고, 에러가 없거나 다중 비트 에러일 경우에는 메모리로부터 읽어 들인 데이터를 그대로 요구 처로 공급한다. 부가적으로 에러 발생 여부와 에러의 종류(single or double)를 외부로 알리는 기능도 갖고 있다.

에러 검사 부분은 데이터와 패리티 코드를 비교하여 에러의 발생 여부 및 에러의 위치 등을 판단하는 기능을 한다. 출력신호 S3~S0가 모두 1이면 에러가 발생치 않은 것이며, 어느 한 비트에라도 0이 있으면 에러가 발생한 것을 말한다. 에러가 발생한 경우 S3~S0의 조합 중 1이 있는 위치가 에러 발생 위치가 된다. 예를 들어 S3~S0이 "1111"의 값을 가지면 에러가 발생한 것이 아니며, "0001"의 값을 가지면 D0에 에러가 발생한 것을 말한다. 에러 발생 통보 부분은 에러 검사 부분의 출력 S3~S0를 해석하여 단일 비트 에러일 경우에는 출력신호 SE를 활성화시키고, 다중 비트 에러일 경우엔 출력신호 DE를 활성화시킨다. 에러가 발생치 않은 경우에는 SE와 DE 모두 비활성화 상태를 유지한다.

그림 3은 에러정정 및 발생부의 일부를 실제 구현한 회로의 일부이다.

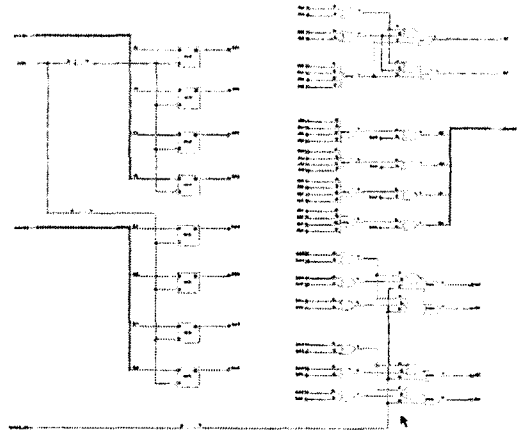


그림 3 에러 정정 및 출력부 회로도
Fig. 3 Error correction and output schematic

2.4 시뮬레이션 결과

시뮬레이션 조건은 12[MHz]의 클럭 소스로 약 2.9[usec] 시간 동안 실행하였다. 프로세서 라이트 신호는 ~CPU_WR이고 리드 신호는 ~CPU_RD이며 메모리(RAM)에 라이트는 ~RAM_WR 신호이다. DX는 프로세서에서 메모리로 보내지는 데이터이고 DY는 메모리에 읽혀지는 데이터 신호이다. 그리고 CB는 패리티 메모리에 저장되는 데이터를 의미한다. 전체 시스템을 리셋 후 약 1.6[usec]의 [write cycle]에서는 프로세서가 데이터 0xDBDB 값과 패리티 데이터 0x2424를 메모리에 라이트가 이루어진다. 그리고 가상으로 패리티 메모리에 1비트 에러가 발생하였다고 보고 강제로 0x2424를 0x2420으로 수정하였다. 그림 5를 보면 약 2.439[usec]에서 1비트 에러 발생 신호 SE가 활성화되는 것을 알 수 있고 2.499 이후에 데이터를 보면 0x2424로 데이터가 수정되어 있다.

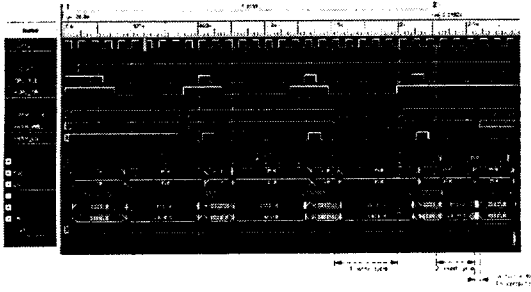


그림 4 1비트 에러 시뮬레이션 전체 결과
Fig. 4 Simulation result for 1bit error

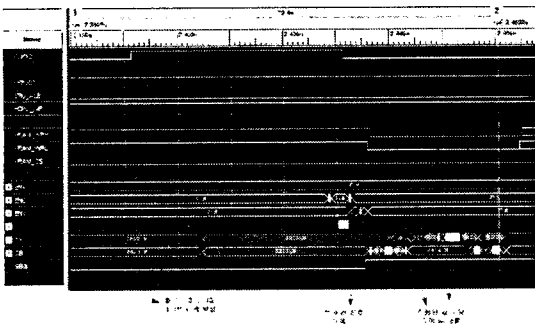


그림 5 1비트 에러 시뮬레이션 수정부분
Fig. 5 Correction simulation for 1bit error

3. 결 론

최근의 전자회로 설계는 반도체 기술의 발달로 여러 개의 IC 소자를 하나의 칩에 넣어 설계할 수 있는 FPGA 등을 많이 사용되고 있다. 직접도가 높은 FPGA 소자 등은 방사선에 대한 영향이 메모리 소자나 IC 소자들과는 달리 높게 나타나기 때문에 이를 이용하여 SRAM 메모리에 발생하는 에러를 검출하고 정정할 수 있는 디지털 회로 설계 기법을 본 논문에서 기술하였고 시뮬레이션을 통하여 검증하였다. 제안된 방법은 데이터 1 바이트를 0, 2, 4, 6번 비트와 1, 3, 5, 7번 비트로 구성된 두개의 니블로 나누어 각 니블에 대하여 독립적으로 EDAC를 수행한 후 결과를 조합하여 다시 1 바이트로 구성하는 방식을 사용하였다. 그러므로 한 워드에 대한 에러 검출 및 정정은 4비트 크기의 그룹 4개 각각에 대한 독립적인 에러 검출 및 정정 작업으로 완료된다. 이러한 알고리즘으로 인하여, 에러가 짝수 비트와 홀수 비트에서 각각 하나씩 발생한 경우에는, 한 바이트에서 발생한 두개의 에러를 검출하여 정정할 수 있게 하였다. 이러한 성능은 시뮬레이션을 통해 한 비트 에러 시 수정된 값을 프로세서에서 공급하고 동시에 메모리에 수정된 값을 자동으로 갱신됨을 확인하였다. 향후 2비트 에러가 발생시 검출하는 시뮬레이션과 1비트 에러 및 정정 및 2비트 에러 검출을 시험을 통해 검증하고자 한다.

[참 고 문 헌]

- [1] 장영근 외, “인공위성 시스템”, 동문사, 1999.
- [2] 김기형 외, “소형 위성의 제어를 위한 컴퓨터 시스템의 설계 및 구현”, J. Astron Space Science 12(2), S52-S66, 1996.
- [3] 김대영 외, “인공위성 핵심기술 선행개발 연구”, 한국항공우주연구원, 2001.
- [4] 박성동 외, “우리별 1, 2호 위성 시스템 개요”, J. Astron Space Science 13(2), S1-S19, 1996.
- [5] 조영호 외, “다목적 실용위성의 전력계 제어장치를 위한 프로세서 모듈 개발”, 대한 전기학회 하계학술대회 논문집, 용평, D. 2999-2222, 2003.
- [6] KOMPSAT-2 Equipment Specification For EPS Control Unit, KARI, 2002.
- [7] KOMPSAT-2 Component Environment Design and Test Specification, 2002.
- [8] Wiley J. Larson and James R. Wertz, Space Mission Analysis and Design Second Edition, Kluwer Academic Publishers, 1992.
- [9] 80386 Data Sheet, Intel, 2000.