

과학기술위성 2호 탑재컴퓨터의 메모리 세정 방안

Memory Scrubbing for On-Board Computer of STSAT-2

유 상 문*
(Sang-Moon Ryu)

Abstract : The OBC(on-board computer) of a satellite which plays a role of the controller for the satellite should be equipped with preventive measures against transient errors caused by SEU(single event upset). Since memory devices are pretty much susceptible to these transient errors, it is essential to protect memory devices against SEU. A common method exploits an error detection and correction code and additional memory devices, combined with periodic memory scrubbing. This paper proposes an effective memory scrubbing scheme for the OBC of STSAT-2. The memory system of the OBC is briefly mentioned and the reliability of the information stored in the memory system is analyzed. The result of the reliability analysis shows that there exist optimal scrubbing periods achieving the maximum reliability for allowed overall scrubbing overhead and they are dependent on the significance of the information stored. These optimal scrubbing periods from a reliability point of view are derived analytically.

Keywords : STSAT-2, on-board computer, transient bit error, SEU(Single Event Upset), scrubbing, reliability

I. 서론

과학기술위성 2호(STSAT-2)는 우리나라 최초의 국내 개발 소형위성발사체(KSLV-1)에 실려 국내에서 첫 번째로 발사될 100kg급 지구저궤도 위성이다. 과학기술위성 1호의 개발 당시 축적된 경험과 기술을 바탕으로 순수 국내 기술로 개발되고 있으며, 한층 향상된 성능과 소형화가 이루어졌다. STSAT-2에서는 지상국으로부터의 명령 및 데이터 처리, 위성 자세 제어, 위성체 운용, 탑재체 운용, 전력부 제어 등의 작업이 탑재 제어기 역할을 하는 탑재 컴퓨터(이하 OBC, On-board Computer)[1]를 중심으로 이루어진다(그림 1).

이러한 OBC 내부의 정보 저장 장치는 우주 환경에서 일시적 오류의 주된 원인인 SEU(Single Event Upset) [2]의 영향을 받게 되며, 이에 대한 대응이 필수적이다.

SEU는 1975년 우주 공간에서 발견된 현상으로, 그림 2 처럼 고에너지 입자가 반도체 부품을 통과할 때 생성되는 전자-정공 쌍에 의해 저장되어 있던 정보의 값이 반전되는 현상을 말한다. 반도체 메모리 부품이 SEU 현상에 가장 큰 영향을 받으며, 그 집적도가 높아짐에 따라 항공기 운항고도 및 지상에서도 SEU 현상이 발견되고 있다[3,4].

STSAT-2의 OBC에는 SEU에 의한 메모리 장치의 일시적 비트 오류에 대한 대응책으로서, 프로그램 실행 코드와 각종 정보가 저장되는 휘발성 메모리(SRAM)의 정보 보호를 위한 오류 검출 및 정정(EDAC, Error Detection and Correction) 회로와 추가의 메모리가 장착되었다. 그리고 일시적 비트 오류가 채용된 EDAC 회로의 정정 능력 범위 이상으로 누적되지 않도록 주기적인 메모리 세정(scrubbing)을

수행하여 준다.

메모리 세정은 CPU에 의해 수행되는 작업으로 세정 주기가 짧아지면 일시적 비트 오류가 EDAC 회로의 정정 능력 범위 이상으로 누적될 가능성이 줄어들지만 CPU의 부담은 증가한다. 이와 반대로 세정 주기가 길어지면 세정에 따른 CPU의 부담은 줄어들지만 일시적 비트 오류가 누적될 가능성이 커져서 OBC의 오동작 또는 동작 중단으로 이

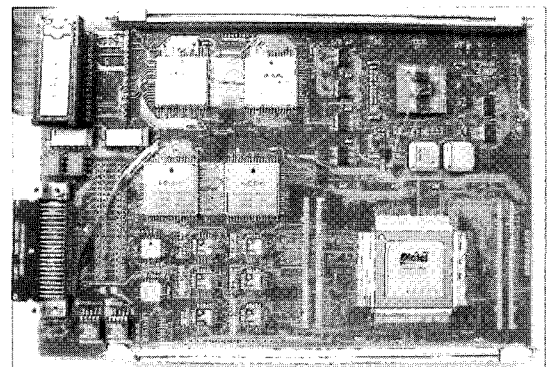


그림 1. STSAT-2 탑재 컴퓨터의 비행 모델.

Fig. 1. Flight model of STSAT-2 OBC.

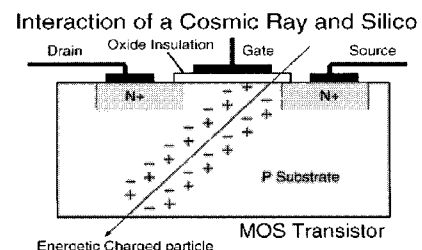


그림 2. SEU의 원리.

Fig. 2. Principle of SEU.

* 책임저자(Corresponding Author)

논문접수 : 2007. 1. 25., 채택확정 : 2007. 2. 17.

유상문 : 군산대학교 전자정보공학부(smryu@kunsan.ac.kr)

※ 본 논문은 한국과학기술원 인공위성연구센터에서 지원하여 연구하였음.

어질 수 있다. 따라서 OBC의 성능 및 요구되는 신뢰도 수준을 함께 고려한 최적의 세정 방안을 찾을 필요성이 있다.

본 논문에서는 STSAT-2 OBC의 메모리 시스템에 대해 간략히 설명하고, EDAC 회로와 주기적 메모리 세정이 적용된 경우에 메모리에 저장되어 있는 정보의 신뢰도를 해석하였다. 그리고 이를 바탕으로 STSAT-2 OBC를 위한 최적의 세정 방안을 제안한다.

II. STSAT-2 OBC의 메모리 시스템

STSAT-2 OBC의 메모리 시스템은 그림 3과 같은 구조를 갖는다. 메모리 시스템은 크게 프로그램 메모리, 램 디스크(RAM disk), 패리티(parity) 메모리로 나눌 수 있으며, 모두 SRAM을 이용하였다. 프로그램 메모리에는 OBC에서 실행되는 자세 제어, 전력 관리, 임무 수행 등의 각종 소프트웨어 태스크와 이들을 위한 운영체제가 상주하며, 램 디스크에는 파일 시스템이 구현되어 있어 STSAT-2의 운영 과정에서 발생하는 정보나 운영에 필요한 정보가 파일 형태로 저장된다.

CPU가 프로그램 메모리와 램 디스크에 기록하거나, 이들로부터 읽어 들이는 정보는 항상 EDAC 회로를 통과하게 되어, SEU에 의해 발생할 수 있는 비트(bit) 오류를 극복하게 된다. CPU가 정보를 기록할 때에는 EDAC 회로가 기록되는 정보에 대응하는 패리티 비트들을 만들어내고, 이 패리티 비트들은 패리티 메모리에 저장된다. CPU가 정보를 읽어낼 때에는 해당 정보를 기록할 때 생성된 패리티 비트들이 함께 읽혀져 EDAC 회로에 의해 비트 오류의 존재 여부가 판단되고, 오류가 발생한 경우 오류 정정 동작이 행해져 CPU는 오류가 정정된 정보를 제공 받게 된다. EDAC 회로는 (7, 4) Hamming 코드[5]를 하드웨어로 구현한 것으로 4비트 당 1비트 오류에 대한 정정이 가능하다. 프로그램 메모리와 램 디스크용으로 각각 2MB의 SRAM이 장착되었으며, 이들을 위한 패리티 저장용으로 4MB의 SRAM이 장착되었다. 패리티 메모리 1은 프로그램 메모리에 저장되는 정보에 대한 패리티를, 패리티 메모리 2는 램 디스크에 저장되는 정보에 대한 패리티를 저장한다.

비트 오류의 정정은 CPU가 오류가 포함되어 있는 워드(Word)를 읽고, 다시 동일 주소에 기입해 주는 과정에서 EDAC 회로에 의해 이루어지며, 어느 주소의 워드에 일시적 오류가 발생할지 알 수 없으므로, CPU가 접근할 수 있

는 모든 메모리 영역에 대해 읽기와 쓰기 동작을 주기적으로 실행해 주어야 한다. 이러한 작업을 메모리 세정이라 하며, 그 실행 주기는 프로그램 메모리, 램 디스크 그리고 패리티 메모리의 각 워드에서 발생할 수 있는 일시적 비트 오류의 누적 발생 개수가 적용된 EDAC 회로의 정정 능력을 벗어나지 않도록 충분히 작게 설정되어야 한다.

이러한 메모리 세정은 별도의 전용 하드웨어를 추가하여 수행할 수도 있지만[6], STSAT-2 OBC에서는 CPU가 직접 수행하도록 설계되어 있다. 소프트웨어에 의해 CPU가 메모리 전 영역을 주기적으로 읽고 써주는 작업을 수행하며, 프로그램 메모리와 램 디스크의 모든 워드를 1회 세정하는데 소요되는 시간이 8초로 측정되었다. 이것은 자세 제어와 전력 관리 등의 중요 태스크를 실행해야 하는 CPU에게 작지 않은 부담으로 작용하게 된다. 메모리 세정에 따른 부담에도 불구하고, 세정 전용의 하드웨어를 장착하지 못한 것은 OBC의 비용, 크기, 소비 전력 등의 부담을 줄이기 위해서이다.

메모리의 각 워드에서 채용된 EDAC 회로에 의해 정정 불가능한 비트 오류가 존재할 확률은 세정 주기에 비례한다. 세정 주기가 커지면 CPU의 부담은 줄어들게 되어 메모리 세정으로 인한 OBC의 성능 저하가 적게되지만, 일시적 비트 오류가 정정되지 않고 누적될 가능성이 커지게 된다. 이와 반대로, 세정 주기가 작아지면 일시적 비트 오류가 누적될 확률은 작아지지만 CPU의 부담이 증가하게 되어 OBC의 성능저하가 커지게 된다. 따라서 메모리에 저장된 정보의 신뢰성과 OBC의 성능을 함께 고려한 적절한 세정 주기를 정하는 것은 중요한 문제이다.

III. STSAT-2 OBC의 메모리 세정 방안

EDAC 회로와 메모리 세정이 적용되는 상황에서 메모리에 저장되어 있는 정보에 대한 신뢰도 해석 결과를 보여주고, STSAT-2 OBC의 프로그램 메모리와 램 디스크 영역을 서로 다른 세정 주기로 세정할 경우, 저장된 정보에 대한 신뢰도를 유도한다. 그리고 이를 바탕으로 허용된 총 세정 부담을 유지하면서 저장된 정보의 신뢰도를 최대화하는 영역별 최적 세정 주기를 유도한다.

1. 메모리 세정에 따른 신뢰도 해석

한 개의 비트에 SEU에 의한 일시적 오류가 평균값 λ 를 갖는 poisson 과정에 따라서 발생하고 주기적 세정이 실행되는 환경에서 일시적 비트 오류에 대한 OBC의 메모리 시스템에 저장된 정보의 신뢰도 함수[7] $R(t)$ 를 구한다. $R(t)$ 는 임의의 시간 $t(>0)$ 에 메모리 시스템을 구성하는 모든 워드들에서 채용된 EDAC 회로에 의해 정정 불가능한 오류가 존재하지 않을 확률로 정의하며, $t=0$ 에서 메모리가 초기화된다면 $R(0)=1$ 이 된다.

신뢰도 함수 유도를 위해 OBC의 메모리 시스템의 총 워드의 수를 M , 하나의 워드 안에서 정보의 저장에 할당된 비트의 수를 w , 패리티 비트의 수를 c , 메모리 세정 주기를 T 라고 한다. 그리고 EDAC 회로는 앞서 2장에서 설명된 것과 같이 하나의 워드($w+c$) 비트) 안에서 한 개의 일시적 비트 오류를 정정할 수 있다고 가정한다.

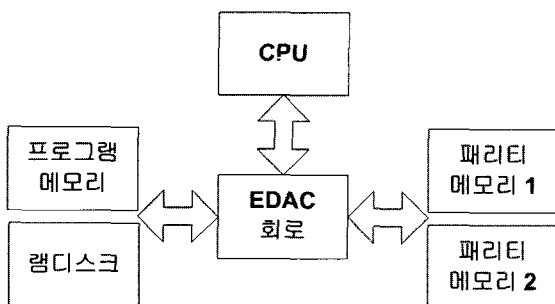


그림 3. STSAT-2 OBC 메모리 시스템.

Fig. 3. Memory system of STSAT-2 OBC.

일시적 오류가 평균값 λ 를 갖는 poisson 과정에 따라서 발생한다고 가정하였으므로, 임의의 시간 구간 $[0, t]$ 동안 비트당 일시적 오류가 발생하지 않을 확률은 $e^{-\lambda t}$ 이며, 오류가 발생할 확률은 $(1 - e^{-\lambda t})$ 이다.

임의의 시간 t 에서 하나의 워드에 정정 불가능한 오류가 존재하지 않기 위해서는 워드를 구성하는 $(w+c)$ 개의 모든 비트들에 일시적 오류가 발생하지 않거나, $(w+c)$ 개의 비트들 중 오직 하나의 비트에서만 일시적 오류가 발행하여야 한다. 따라서 한 개의 워드에 대한 정보의 신뢰도 함수 $r_o(t)$ 는 다음과 같다.

$$r_o(t) = e^{-\lambda(w+c)t} + (w+c)(1 - e^{-\lambda t})e^{-\lambda(w+c-1)t} \approx 1 - \frac{1}{2}(w+c)(w+c-1)\lambda^2 t^2 \quad (1)$$

실제 λ 의 값은 매우 작기 ($< 10^{-9}$) 때문에 지수 함수를 테일러 전개한 후 우세항만을 사용하여 근사화하였다.

그리고 M 개의 워드로 구성된 메모리 시스템에 정정 불가능한 오류가 존재하지 않기 위해서는 모든 M 개의 워드가 정정 불가능한 오류를 갖고 있지 않아야 한다. 따라서 메모리 시스템에 저장된 정보의 신뢰도 함수 $R_o(t)$ 는 다음과 같다.

$$R_o(t) = [r_o(t)]^M \approx 1 - \frac{M}{2}(w+c)(w+c-1)\lambda^2 t^2 \quad (2)$$

(1)의 경우와 마찬가지로 λ 의 값은 매우 작기 때문에 이항 전개하여 그 우세항만을 사용하여 근사화하였다.

위의 $R_o(t)$ 는 세정이 적용되지 않을 때의 신뢰도 함수이다. 세정이 적용되면 세정 주기 T 마다 정정 가능한 범위 내의 일시적 비트 오류들이 정정되므로, 세정이 적용된 메모리 시스템에 저장된 정보의 신뢰도 함수 $R(t)$ 는 매 세정 작업이 완료된 시점에서 모든 비트 오류가 정정되고, 최근 세정 작업 이후에 정정 불가능한 비트 오류가 발생하지 않을 확률과 같다. 이것은 다음과 같이 표현될 수 있다.

$$R(t) = [R_o(T)]^n R_o(\tau) = (1 - CT^2)^n (1 - C\tau^2), t \geq 0 \quad (3)$$

여기서, $C = \frac{M}{2}(w+c)(w+c-1)\lambda^2$, $t = nT + \tau$ 이고, τ 는 $0 \leq \tau < T$ 를 만족하는 실수이며, n 는 $n \geq 0$ 를 만족하는 정수이다.

그림 4는 동일 파라미터를 사용한 경우의 $R_o(t)$ 와 $R(t)$ 를 함께 보여 준다. 세정 작업이 실행되지 않으면 일시적 비트 오류가 계속 누적 발생되어 신뢰도 함수가 지수함수적으로 감소하고, 주기적 세정 작업이 실행되면 매 세정 주기마다 지수함수적 감소가 초기화되어 신뢰도 함수의 하향 추세가 현격히 둔화되는 것을 알 수 있다. 그리고 (3)의 신뢰도 함수 $R(t)$ 는 다음과 같이 근사화할 수 있다.

$$R(t) \approx (1 - CT^2)^{\frac{t}{T}}, t \geq 0 \quad (4)$$

신뢰도 함수와 함께 신뢰도의 지표로 자주 사용되는 것이 MTTF(Mean-Time-to-Failure)[7]이며, 신뢰도 함수 $R(t)$ 를

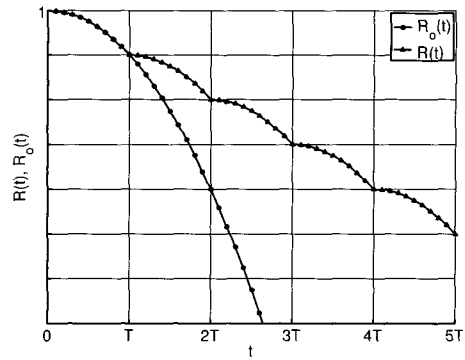


그림 4. 신뢰도 함수 비교.

Fig. 4. Comparison of reliability functions.

이용하여 다음과 같이 정의된다.

$$MTTF = \int_0^{\infty} R(t) dt \quad (5)$$

(5)와 같이 정의되는 MTTF는 메모리 시스템에 정정 불가능한 오류가 발생할 때까지의 평균 시간을 의미하며, 근사화된 $R(t)$ 를 이용하여 MTTF를 다음과 같이 구할 수 있다.

$$MTTF \approx -\frac{T}{\ln(1 - CT^2)} \quad (6)$$

2. STSAT-2 OBC에서의 신뢰도

램 디스크에는 위성의 운영 과정 중 생성된 데이터가 파일 형태로 저장되며, 프로그램 메모리에서는 위성의 생존 및 임무 수행에 필수적인 자세 제어, 전력 관리, 상태 관리 소프트웨어 태스크 등이 실행된다. 따라서 프로그램 메모리에 저장되어 있는 정보의 중요도가 램 디스크에 저장되어 있는 정보의 중요도 보다 현격히 크다고 할 수 있다.

메모리 시스템을 구성하는 두개의 영역이 서로 다른 정보 중요도를 갖고 있으므로, 두 영역을 모두 세정하는데 필요한 총 세정 부담을 동일하게 유지하면서, 두 영역을 서로 상이한 주기로 세정하는 것이 두 영역을 서로 동일한 주기로 세정하는 것보다 전체 정보에 대한 높은 신뢰도를 제공할 수 있다. 즉 OBC에 허용된 총 세정 부담을 유지하면서, 프로그램 메모리의 세정 빈도를 높여주고 램 디스크의 세정 빈도를 낮추어 주어 OBC 메모리에 저장된 전체 정보에 대한 신뢰도를 높여줄 수 있는 것이다.

프로그램 메모리와 램 디스크에 서로 다른 세정 주기가 적용된 경우의 신뢰도 해석을 위해 프로그램 메모리와 램 디스크의 총 워드의 수를 각각 M_p 와 M_d 라 하고, 그 세정 주기를 각각 T_p 와 T_d 라 정한다. 그리고 프로그램 메모리에 저장되어 있는 정보에 대해 램 디스크에 저장된 정보의 상대적 비중요성을 나타내기 위해 비중요성 상수 α 를 정의한다. α 는 1보다 큰 실수이며, α 가 클수록 램 디스크에 저장되어 있는 정보의 중요도가 프로그램 메모리의 그것에 비해 떨어진다는 것이며, 이 값은 위성의 발사 이전에 결정되고 운영 중에 갱신되어야 한다.

새롭게 정의된 변수들을 (4)에 적용하면, 프로그램 메모

리를 T_p 의 주기로 세정할 때의 저장된 정보에 대한 신뢰도 함수 $R_p(t)$ 는 다음과 같다.

$$R_p(t) \approx (1 - C_p T_p^\alpha)^{\frac{t}{T_p}}, t \geq 0 \quad (7)$$

여기서 $C_p = \frac{M_p}{2}(w+c)(w+c-1)\lambda^2$ 이다.

3.1절의 신뢰도 해석 모델에서 램 디스크에서의 평균 비트 오류 발생률을 λ/α 로 정의하면 램 디스크에 저장된 정보에 대한 상대적 비중요성을 모델링할 수 있다. 따라서 램 디스크를 T_d 의 주기로 세정할 때의 저장된 정보에 대한 신뢰도 함수 $R_d(t)$ 를 $R_p(t)$ 의 경우와 마찬가지로 구하면 다음과 같다.

$$R_d(t) \approx (1 - C_d T_d^\alpha)^{\frac{t}{T_d}}, t \geq 0 \quad (8)$$

여기서 $C_d = \frac{M_d}{2}(w+c)(w+c-1)\left(\frac{\lambda}{\alpha}\right)^2$ 이다.

전체 정보에 대한 신뢰도 함수는 각 영역에 저장된 정보의 신뢰도 함수의 곱으로 표현되므로 $R_p(t)R_d(t)$ 가 되고, 메모리에 정정 불가능한 오류가 발생할 때까지의 평균 시간을 의미하는 MTTF는 다음처럼 구해진다.

$$MTTF = \int_0^\infty R_p(t)R_d(t)dt \approx \frac{-T_d T_p}{T_d \ln(1 - C_p T_p^\alpha) + T_p \ln(1 - C_d T_d^\alpha)} \quad (9)$$

3. STSAT-2 OBC를 위한 세정 주기

메모리의 세정에 따른 부담은 세정 대상 메모리 영역의 총 워드 수에 비례하고 세정 주기에 반비례하므로 프로그램 메모리와 램 디스크 영역을 동일 주기 T 로 세정할 때의 총 세정 부담은 $\frac{M_p + M_d}{T}$ 로, 각각 다른 주기 T_p 와 T_d 로 세정할 때의 총 세정 부담은 $\frac{M_p}{T_p} + \frac{M_d}{T_d}$ 로 나타낼 수 있다. 따라서 두 영역을 서로 다른 주기 T_p 와 T_d 로 세정할 때의 세정 부담이 동일 주기 T 로 세정할 때의 세정 부담과 같기 위해서는 다음을 만족하여야 한다.

$$\frac{M_p}{T_p} + \frac{M_d}{T_d} = \frac{M_p + M_d}{T} \quad (10)$$

그림 5, 6과 7은 세정 부담이 일정하게 유지될 때 램 디스크에 저장된 정보의 비중요성 상수 α 의 값에 따른 T_p 와 T_d 에 대한 MTTF의 그래프이다. 여기에는 2장에서 소개된 STSAT-2 OBC의 메모리 시스템을 적용하여 $w=4, c=3$ 이다. 그리고 $M_d=M_p=2\text{MB}$ 가 적용되었으며, $\lambda=2 \times 10^{-10}$ 과 $T=500$ 을 가정하였다.

이들로부터 $\alpha=1$ 인 경우에는 $T_d=T_p=T$ 를 만족하는 것이, $\alpha>1$ 인 경우에는 $T_d>T_p$ 를 만족하는 것이 바람직함을 알 수 있다. 즉 램 디스크에 저장된 정보의 중요성이 프로그램 메모리의 그것보다 작은 경우에는 프로그램 메모리를

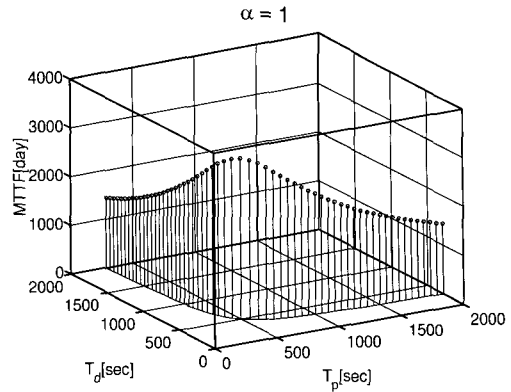


그림 5. T_p 와 T_d vs. MTTF ($\alpha=1$).
Fig. 5. T_p and T_d vs. MTTF ($\alpha=1$).

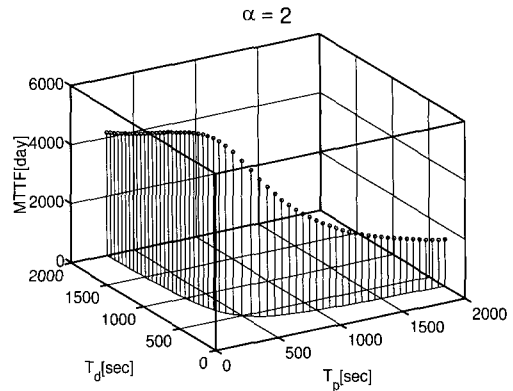


그림 6. T_p 와 T_d vs. MTTF ($\alpha=2$).
Fig. 6. T_p and T_d vs. MTTF ($\alpha=2$).

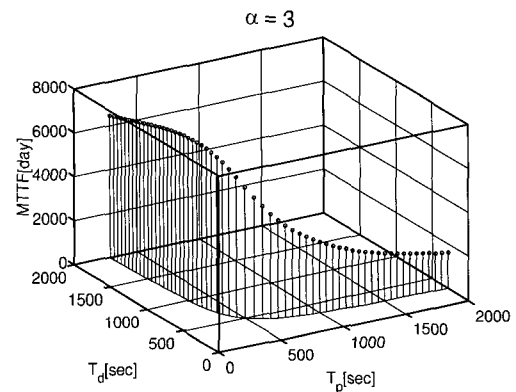


그림 7. T_p 와 T_d vs. MTTF ($\alpha=3$).
Fig. 7. T_p and T_d vs. MTTF ($\alpha=3$).

램 디스크 보다 빈번하게 세정하고, 이로 인해 증가된 세정 부담을 상쇄하기 위해 램 디스크의 세정 빈도를 낮추어주는 것이 신뢰도 측면에서 유리하다는 것이다. 그리고 비중요성 상수 α 와 허용된 세정 부담을 나타내는 T 에 따라 MTTF를 최대로 만들어 주는 T_p 와 T_d 값이 존재함을 알 수 있다.

두 영역을 동일 주기 T 로 세정하는 것과 동일한 세정 부담을 유지하면서, 저장된 정보의 신뢰도를 최대화할 수

있는 각 영역의 세정 주기 T_p 와 T_d 는 (9)와 (10)을 이용해 구성되는 다음의 최적화 문제를 풀면 구할 수 있다.

$$\max \frac{-T_d T_p}{T_d \ln(1 - C_p T_p^2) + T_p \ln(1 - C_d T_d^2)} \quad (11)$$

$$\text{s.t. } \frac{M_p}{T_p} + \frac{M_d}{T_d} = \frac{M_p + M_d}{T} \quad (12)$$

(11)은 메모리에 저장되어 있는 정보의 신뢰도를 최대화 하기 위해 (9)의 MTTF를 최대로 만들어 주는 것을 의미하며, (12)는 두 영역에 대한 세정 부담의 합이 허용된 값으로 유지되어야 하는 제약을 의미한다. $0 < x \ll 1$ 에 대해서 $\ln(1-x) \approx -x$ 이므로 (11)은 다음과 같이 표현될 수 있다.

$$\min C_p T_p + C_d T_d \quad (13)$$

Lagrange 정리[8]에 따르면 다음의 최적화 문제

$$\min f(\mathbf{x}) \quad (14)$$

$$\text{s.t. } h(\mathbf{x}) = 0 \quad (15)$$

에 대해, $h(\mathbf{x}) = 0$ 을 만족하고 $f(\mathbf{x})$ 를 최소화하는 국부 최소점(local minimizer) \mathbf{x}^* 가 존재한다면,

$$\nabla f(\mathbf{x}^*) + \kappa \nabla h(\mathbf{x}^*) = 0 \quad (16)$$

을 만족하는 κ 가 존재하게 된다. 여기서 ∇ 는 Gradient이며, κ 를 Lagrange multiplier라고 한다.

(12)와 (13)을 (14)와 (15)에 대응 시킨 후 Lagrange 정리를 적용하면 다음의 식들이 얻어진다.

$$C_p \left(M_p - \frac{T_p (M_p + M_d)}{T} \right) = 0 \quad (17)$$

$$C_d \left(M_d - \frac{T_d (M_p + M_d)}{T} \right) = 0 \quad (18)$$

(12), (17)과 (18)로부터

$$C_p \left(\frac{M_p + M_d}{T} \right)^2 T_p^2 - 2C_p M_p \left(\frac{M_p + M_d}{T} \right) T_p + M_p (C_p M_p - C_d M_d) = 0 \quad (19)$$

를 얻을 수 있으며, (12)와 (19)를 이용하여 T_p 와 T_d 의 최적값을 구할 수 있다.

그림 7의 경우에 적용된 파라미터에 대해 T_p 와 T_d 의 최적값을 구하면 각각 333과 1002가 얻어지며, 이들은 그림 7의 그래프에서 MTTF가 최대가 되는 T_p 와 T_d 의 값과 일치한다. 구해진 최적값에 대응하는 MTTF와 신뢰도 함수를 T_p 와 T_d 가 모두 500인 경우와 비교하면 표 1과 그림 8과 같다.

두 영역을 동일 주기 500초로 세정하면 저장된 정보의 신뢰도 지표인 MTTF가 5913일로 얻어지지만, 이와 동일한 세정 부담을 유지하면서 프로그램 메모리는 333초마다, 램 디스크는 1002초마다 세정을 하면 MTTF가 7391일로 증가되어 저장된 정보에 대한 신뢰도가 개선된다. MTTF 7391

표 1. 세정 주기에 따른 MTTF 비교.

Table 1. Comparison of MTTF according to scrubbing periods.

T_p [sec]	T_d [sec]	MTTF[day]
500	500	5913
333	1002	7391

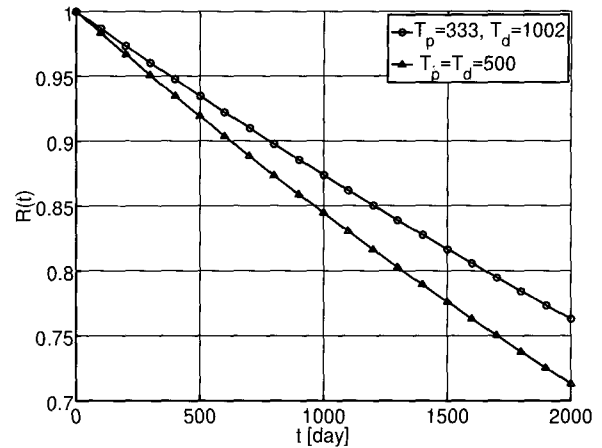


그림 8. 세정 주기에 따른 신뢰도 함수 비교.

Fig. 8. Comparison of reliability functions according to scrubbing periods.

일은 두 영역을 동일 주기 400초로 세정할 때 얻을 수 있는 수치이다. 따라서 전체 세정 부담은 유지하면서 프로그램 메모리와 램 디스크에 서로 다른 최적 세정 주기를 적용하여 저장된 정보에 대한 신뢰도를 개선할 수 있음을 알 수 있다.

신뢰도 함수 $R_p(t)$ 과 $R_d(t)$ 는 단조 감소 함수이므로 이들의 곱인 $R_p(t)R_d(t)$ 역시 단조 감소 함수이다. 따라서 (9)의 MTTF를 최대로 하여주는 T_p 와 T_d 는 임의의 시간 $t(>0)$ 에서의 $R_p(t)R_d(t)$ 를 최대화 하여준다.

프로그램 메모리와 램 디스크에 실제 저장되어 있는 정보의 양은 위성의 운영 상황에 따라 달라질 수 있다. 그러므로 제안된 최적 세정 주기 유도 방법에서 M_p 와 M_d 에 실제 저장되어 있는 정보의 양을 대입하면 좀 더 최적화된 세정 주기를 구할 수 있다.

IV. 결론

위성에서 사용되는 컴퓨터 제어 시스템에서는 메모리 부품이 SEU에 의한 일시적 비트 오류에 가장 취약하며, 이를 위한 보호책이 필수적으로 제공되어야 한다. 과학기술위성 2호(STSAT-2)의 탑재컴퓨터(OBC)에는 4비트 당 1비트의 일시적 오류를 정정할 수 있는 오류 검출 및 정정(EDAC) 기능이 구현되어 있고, 오류의 정정은 CPU가 메모리의 데이터를 읽어 들이는 과정에서 수행되고, 정정된 데이터를 메모리의 동일 주소에 기입하여 주어야 오류가 사라지게 된다. 그리고 CPU가 빈번하게 접근하지 않는 워드에는 일시적 오류가 누적되어 발생할 수 있으므로 주기적으로 메모리의 모든 워드를 읽고 다시 기입해 주는 세정 작업을 실

행해야한다. 이러한 세정 작업은 CPU에겐 부담으로 작용하며, 메모리에 저장된 정보의 신뢰도를 고려하여 적당한 세정 주기를 결정해야 한다.

본 논문에서는 EDAC 회로와 메모리 세정 작업에 의해 보호되는 메모리 시스템에 저장된 정보의 신뢰도를 해석하고, 이를 바탕으로 과학기술위성 2호의 탑재 컴퓨터에 적용할 수 있는 최적 세정 방안을 제안하였다. 제안된 방안을 적용하면 허용된 세정 부담 내에서 메모리 시스템에 저장된 정보의 신뢰도를 최대로 할 수 있다.

참고문헌

- [1] 유창완, 남명룡, 임종태, “과학기술위성 2호 탑재 컴퓨터의 EM 개발 및 구현,” 한국항공우주학회지, 제 34 권, 제 2 호, pp. 101-105, 2, 2006.
- [2] R. H.-Sorensen, E. Daly, F. Teston, H. Schweitzer, R. Nartallo, P. Perol, F. Vandebussche, H. Dzitko, and J. Cretolle, “Observation and analysis of single event effects on-board the SOHO satellite,” *IEEE Trans. Nuclear Science*, vol. 49, no. 3, pp. 1345-1350, Jun. 2002.
- [3] A. Taber and E. Normand, “Single event upset in avionics,” *IEEE Trans. Nuclear Science*, vol. 40, no. 2, pp. 120-126, Apr. 1993.
- [4] E. Normand, “Single event upset at ground level,” *IEEE Trans. Nuclear Science*, vol. 43, no. 6, pp. 2742-2750, Dec. 1996.
- [5] R. W. Hamming, *Coding and Information Theory*, Prentice-Hall, 1980.
- [6] 김대영, 조창범, 강석주, 채태병, “하드웨어 메모리 스크러버 설계,” 항공우주기술, 제 2 권, 제 1 호, pp. 73-79, 2, 2003.
- [7] B. W. Johnson, *Design and Analysis of Fault-Tolerant Digital Systems*, Addison-Wesley, 1989.
- [8] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*, John Wiley & Sons, 2001.

유 상 문



1970년 1월 11일생. 1992년 금오공과대학교 전자공학과 졸업. 1995년 한국과학기술원 전기 및 전자공학과 석사. 2006년 동 대학원 박사. 1995년~2000년 LG전자(주). 2000년~2004년 한국과학기술원 인공위성연구센터. 2006년~

현재 군산대학교 전자정보공학부 교수. 관심분야는 임베디드 제어 시스템, 내고장성 시스템 및 실시간 제어 시스템.