결함허용 시스템의 하드웨어 여분구조에 대한 연구

A study on Hardware Redundancy Architecture of Fault-Tolerant System

신덕호* 이종우* 이재호* 이기서**
Ducko shin Jong-woo LEE Jae-ho LEE Key-soe LEE

ABSTRACT

This paper is to discuss the hardware redundancy architecture of fault-tolerance system with using redundancy. Each architecture will be studied to implement fault-tolerance in classifying hardware redundancy architecture as passive, active and hybrid hardware redundancy. Therefore Fault-Masking and Fault-Detecting Techniques in each redundancy architecture is studied.

1. 서 론

1.1 결함허용의 목표

가장 일반적으로 가질 수 있는 의문은 왜 결함허용이 중요하며 많은 엔지니어들이 관심을 갖는가이다. 결함허용은 시스템에서 발생할 수 있는 결함을 예측하여, 결함에 대한 대처방안을 설계에 적용하고 시스템의 신뢰도, 가용도 등의 설계목표를 만족시킨다. 시스템의 설계요구사항 중 대표적인 것은 신뢰성, 가용성, 안전성, 성능성(Performability), 독립성, 유지보수성 그리고 테스트의 용이성이며 결함허용은 이러한 요구사항을 이행할 수 있는 능력이다.

1.2 여분의 개념

결함허용설계에서 물리적 하드웨어 여분을 사용한 여분의 개념은 오래 전부터 사용되었다. 결함허용에 사용되는 가장 일반적인 방법은 시스템의 하드웨어를 모듈단위 또는 시스템 단위로 다중화 하는 방법이다. 최근에는 여분에 대한 좀더 구체적인 활용방법과 여분을 사용하여 시스템의 결함허용 능력을 증대시키는 연구가 활발하게 진행되고 있다. 여분은 시스템이 정상 동작하는데 필요한 요소 외에 정보, 자원, 시간을 추가하는 것으로 정의한다. 또한 여분은 다음과 같이 분류할 수 있다.

* 한국철도기술연구원, 정회원
** 광운대학교 정교수, 정회원

a. 하드웨어 여분(Hardware Redundancy)은 결함을 검출하고 허용하기 위해 추가적인 하드웨어를 사용하는 것.
b. 소프트웨어 여분(Software Redundancy)은 결함을 검출하고 허용하는 것을 가능하게 하기 위해 추가적인 소프트웨어를 사용하는 것.
c. 정보여분(Information Redundancy)은 오류검출코드와 같이 주어진 기능을 수행하기 위해 코드의 추가 또는 정형화된 규칙을 통해 정보를 가공하는 것.
d. 시간여분(Time Redundancy)은 시스템의 결함검출과 결함의 허용을 위해 추가적인 시간을 사용하여 것.

2. 본 론

2.1 하드웨어여분

하드웨어여분은 물리적으로 같은 모듈을 중복해서 사용하는 방법으로 디지털 시스템에서 가장 많이 사용되는 방법이다. 시스템의 전자화로 시스템의 물리적인 크기가 작아지고 가격이 저렴해지면서 가장 유용한 방법으로 선택되고 있다. 하드웨어여분에는 수동방식(Passive), 능동방식(Active) 그리고 혼합방식(Hybrid)이 있다.

수동방식은 결함은폐(Fault Masking)를 개념으로 설계할 때 사용되며, 결함발생을 은폐하고 결함이 오류로 발전하는 것을 방지한다. 수동기법의 설계는 시스템 측면에서 별도의 동작을 요구하지 않으며 결함에 대하여 검출보다는 은폐를 목적으로 한다.

능동방식은 다이내믹 방법이라고도 불리며 결함을 검출하고 결함이 발생한 하드웨어를 시스템에서 적극적으로 제외시켜 결함허용을 수행하는 방식이다. 다시 말하면 결함을 허용하기 위한 재구성을 의미한다. 능동하드웨어 여분은 결함검출(Fault Detection), 결함격리(Fault Location), 결함회복(Fault Recovery)을 수행하여 결함허용을 수행한다.

혼합방식은 수동방식과 능동방식을 모두 사용하는 방법으로 수동방식의 결함은폐와 능동방식에서의 결함검출, 결함격리, 결함회복을 함께 사용하여 하드웨어 결함을 제거하고 대체 함으로써 결함허용을 구현한다.

결론적으로 여분(Redundancy)은 시스템에 부수적인 요소를 의미하며, 다른 요소에서 결함이 발생하기 전까지는 필요하지 않다가 결함이 발생하면 발생된 부분을 대체하기 위해 사용된다. 따라서 혼합방식은 중요한 연산(Critical Computation)응용에서 순간적인 오류의 방지와 고신뢰성을 구현하기 위해 가장 많이 사용되며, 여분을 사용한 시스템에서 구현비용이 가장 높다.

2.1.1 수동하드웨어 여분(Passive Hardware Redundancy)

수동하드웨어 여분은 결함발생을 은폐하기 위해 보트(Vote)메커니즘을 사용한다. 수동방식에서 가장 많이 사용되는 방법은 다수결 보팅(Majority Voting)이다. 보터를 사용한 수동여분구조는 결함허용을 위해 결함의 검출이나 시스템의 재구성을 요구하지 않으므로, 결함에 대한 허용을 충실히 따른다고 할 수 있다.

수동하드웨어 여분에서 가장 일반적인 방법은 TMR(Triple Modular Redundancy)이다. TMR(그림1)의 기본개념은 하드웨어를 3중화시키고 다수결 보터가 시스템의 출력을 결정하는 방식이다. 만약 하나의 모듈에서 결함이 발생해도 나머지 두 정상모듈(결함이 발생하지 않은 모듈)의 결과가 다수결 보터의 동작으로 결함이 발생한 모듈의 결과를 은폐하는 방식이다.

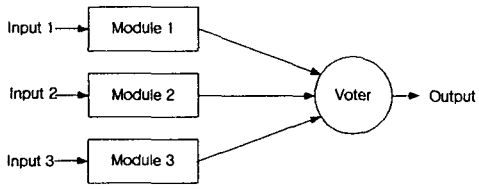

그림1. 다수결 보터를 사용하여 단일 출력을 발생하는 TMR시스템(Triple modular redundancy)

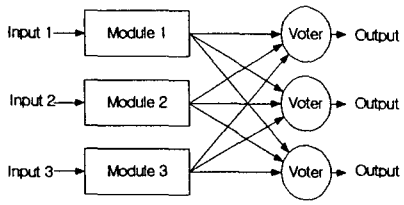

그림2. 보터의 고장을 고려한 3중 보터를 사용한 TMR시스템

TMR시스템의 하드웨어는 프로세서, 메모리 또는 기타자원들이 다중화 모듈로 구현되며, 각각의 다중화 모듈에는 존재하는 결함을 검출하기 위해 동일한 기능을 수행하는 소프트웨어를 모듈에 포팅하여 결과를 보팅한다.

TMR에서 가장 어려운 부분은 보터(Voter)이다. 만약 보터에서 결함이 발생하면 시스템 전체로 결함이 확산되어 시스템전체가 고장난다. 즉, TMR시스템의 신뢰도는 보터의 신뢰도보다 높을 수 없다. 이렇게 단일 요소의 고장이 시스템 전체의 고장을 발생시키는 것을 Single Point of Failure 라고 한다. 보터의 고장에 대한 영향을 방지하기 위한 방법 중에 하나가 그림2와 같이 보터를 3중화시켜 각각 독립적인 출력을 생성하는 것이다. 그림2에서 3개의 기능모듈은 각각의 입력을 받아서 각자 기능을 수행하고 기능모듈들의 출력이 각각 보팅된다.

보터 각각의 출력은 결함이 발생하지 않으면 모두 동일한 값을 출력하며, 그림2와 같은 설계를 그림3과 같이 TMR시스템 내부에서 응용할 수 있다. 만약 시스템 내부의 보터가 고장나면 다음단계의 보터에서 고장이 검출된다. 따라서 다음 단계의 출력에서는 이전단계의 고장을 포함한 오류결과를 올바른 값으로 정정하여 출력한다.

단일 고장입력에도 불구하고 올바른 3개의 출력을 발생하므로, 보터를 3중화한 TMR시스템을 복구기관(Restoring Organ)이라고 한다. 그림3과 같은 TMR 구조는 보터를 다중화 하여 오류가 없는 신호로 복원이 가능하다.

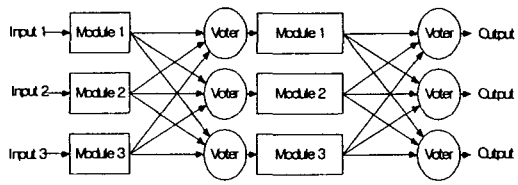

그림3. 각 단계의 보터결함을 억제하는 다단계 TMR 시스템의 구조

하드웨어 보팅 또는 소프트웨어 보팅을 선택하는데는 다음의 요소가 많은 비중을 차지한다.

a. 보팅을 수행하기 위한 프로세서의 가용도
b. 보팅이 수행해야할 최대 지원속도
c. 시스템에서 수용할 수 있는 공간, 전력소모량 및 중량한계
d. 소요되는 보터의 수
e. 시스템의 설계변경시의 보터의 유연성

단일 보터구조의 TMR시스템에서 보터의 결함을 허용하기 위한 방법으로 그림4의 Flux-Summing 기법을 들 수 있다. 그림4의 TMR시스템은 전류를 제어하여 계전기를 구동하는데 적용할 수 있다. Flux-Summing 기법은 시스템을 폐루프로 구현하여 결함을 보상할 수 있다.

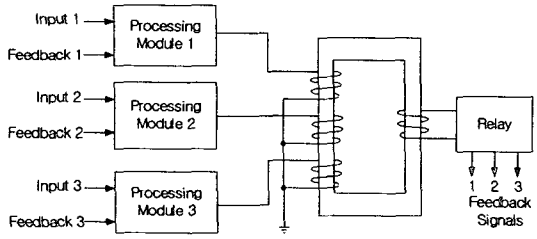

그림4. Flux-summing 방식을 사용한 결함은폐 기능을 갖는 TMR시스템의 계전기 출력

그림4에서 Flux-Summing 트랜스포머는 1차측에 3번, 2차측에 1번의 권선으로 구성되었다. 따라서 2차측에 발생되는 전류는 1차측의 3가지 권선들의 전류합이된다. 따라서 결함이 발생하지 않은 상태에서는 각각의 모듈이 2차측에서 발생하는 전류의 1/3씩을 공급하며, 모듈이 고장나면 센서에 의해 검출되어 고장난 모듈은 전류공급을 중단한다.

고장난 모듈이 차단되면 계전기는 현재 위치를 유지하는데 필요한 전류의 1/3을 잃게되므로, 남아있는 두 모듈은 궤환 센서에 의해 계전기 제어에 필요한 전류를 두 모듈이 공급하여 계전기의 상태를 유지시킨다.

Flux-Summing을 사용한 계전기 제어에서 예측할 수 있는 또 하나의 고장시나리오로 고장난 모듈이 최대 전류를 공급하는 경우를 들 수 있다. 이러한 상황이 발생하면 궤환된 값에 의해 나머지 정상모듈의 전류량이 변화한다. 고장에 의해 최대 전류 량이 발생하는 경우 정상모듈들이 반대극성으로 전류를 공급하여 상쇄시키므로 계전기의 제어전류를 제어한다. Flux-Summing 기법이 보팅은 아니지만 결함은폐(Fault Masking)와 같은 효과를 낸다는 것이다.

2.1.2 능동하드웨어 여분(Active Hardware Redundancy)

능동하드웨어 여분기법은 결함검출, 결함격리, 결함회복을 사용하여 결함허용을 구현한다. 많은 분야의 설계에서 결함의 검출은, 발생된 결함에 의해 오류가 발생한 것으로 결함의 발생여부를 판단하므로 오류검출(Error Detection), 오류격리(Error Location), 오류회복(Error Recovery)이라고 표현한다.

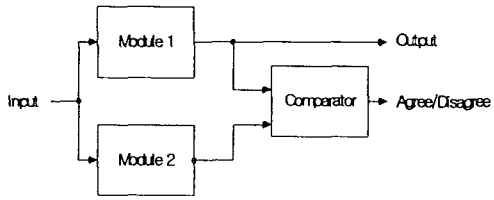

그림5. 비교기를 사용한 이중화 (Duplication with Comparison)

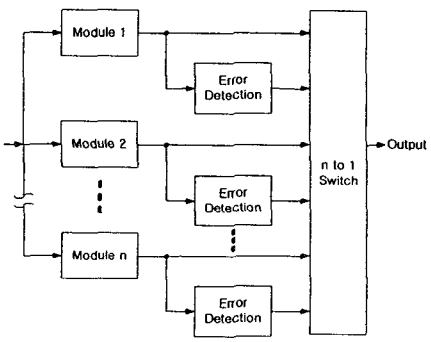

그림6. 대기여분의 구조(Standby Sparing)

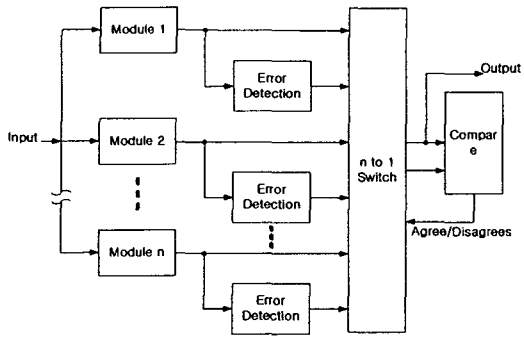

그림7. Pair-and-a-Spare 기법

비교기를 사용하거나 자기검사회로(Self Checking Logic)를 사용하여 오류로 발전한 결함을 검출하는 방식을 사용하므로, 발생된 결함의 억제보다는 시스템 재구성을 통하여 결함허용을 구현한다. 이러한 능동 하드웨어 여분은 다음의 그림들과 구조로 구현할 수 있다.

2.1.3 혼합하드웨어 여분 (Hybrid Hardware Redundancy)

혼합하드웨어 여분의 기본개념은 능동과 수동 하드웨어 여분의 장점만을 조합한 것이다. 결함은폐는 시스템에서 오류를 유발하는 것을 방지하고 결함검출, 결함격리, 결함회복은 결함이 시스템에서 발생하는 경우 시스템 재구성을 위해 사용된다. 혼합 하드웨어 여분으로 시스템을 구성하기 위해서는 많은 하드웨어들이 요구된다. 따라서 혼합여분구조는 고신뢰성과 고가용성이 기대되는 응용분야에서 널리 사용된다.

혼합여분구조의 예로는 N-모듈 여분구조, 자체정화 여분구조(Self-Purging Redundancy) 및 Triple-Duplex Architecture를 들 수 있다. 이 중 이중계로 구성된 모듈을 세 개 사용하여 구성한 Triple-Duplex구조는 결함을 은폐하는 수동여분구조의 장점과 이중으로 구성된 모듈을 비교회로를 사용하여 결함을 검출하는 능동여분구조의 장점을 모두 가지고 있다.

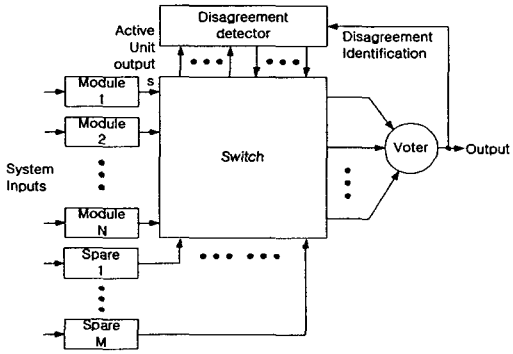

그림8. 여분을 갖는 N-모듈 여분구조 (N-Modular Redundancy with Spare)

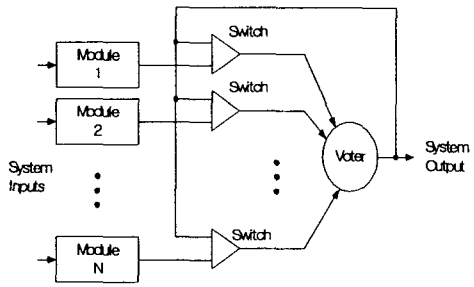

그림9. 자체정화 여분구조 (Self-Purging Redundancy)

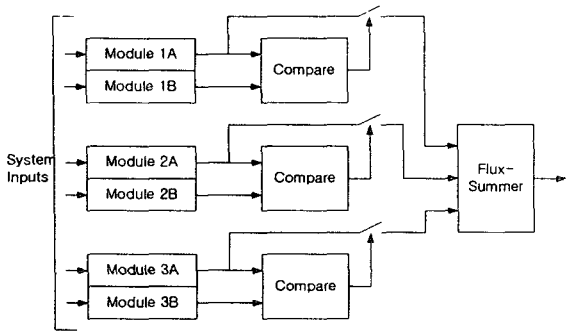

그림10. Triple-Duplex Architecture

그림10은 Flux-Summing Arrangement를 보팅 메커니즘으로 사용한 Triple-Duplex 구조를 예로든 것이다.

3. 결 론

본 논문에서는 하드웨어 여분구조의 세 가지 기본형태에 대하여 연구하였으며, 능동, 수동, 혼합하드웨어 여분구조의 장단점을 도출하였다. 각각이 여분구조에 대한 차이점을 정리하면 다음과 같다.

a. 수동기법은 결함은폐를 기본으로 한다,
b. 능동기법은 결함은폐는 할 수 없지만, 결함검출, 격리, 그리고 복구기법을 사용하여 시스템을 재구성한다.
c. 혼합기법은 결함은폐와 재구성을 모두 가능하게 한다.

따라서 위와 같은 여분구조를 응용분야에 사용하기 위해서는 시스템의 최종 설계목표가 신뢰성인지 가용성인지를 결정하여, 높은 신뢰도가 요구되는 경우에는 수동여분구조를 적용하고, 높은 가용도가 요구되는 경우에는 능동여분구조를 적용한다. 또한 시스템 설계 제약조건을 고려하여 높은 가용도와 신뢰도를 동시에 만족시키는 혼합여분구조도 사용한다.

따라서 철도신호제품의 많은 비중을 차지하는 디지털 제어시스템의 설계시에도 적용대상에 대한 설계목표의 정확한 분석을 통하여 여분구조가 선택되어야 한다.

참고문헌

[1] "Design and Analysis of Fault-Tolerant Digital Systems" written by Barry W. Johnson Edited by Addison-Wesley.1989.
[2] "Fault-Tolerant and Fault Testable Hardware Design" written by parag K. Lala. 1985.
[3] "Fail-Safe Inteface for VLSI : Theoretical Foundations and Implementation" Michael Nicolaidis, Member, IEEE Computer Society. Vol. 47. No. 1 JAN. 1998.