

고장 검출 버스를 이용한 고장 감내 시스템 설계

정우석*, 송광석*, 이광선° **, 신진욱**, 박동선**

전자통신연구원*

전북대학교 정보통신공학과**

Email : ray@multilab.chonbuk.ac.kr

Design of a Fault Tolerant System Employing Fault Detection Bus

Woo-Sug Jung*, Kwang-Suk Song*

Kwang-Sun Lee° **, Jin-Wook Shin**, Dong-Sun Park**

Electronics and Telecommunication Research Institute*

Dept. of Information and Communication Engineering, Chonbuk National Univ.**

Abstract

A fault-tolerant system should have a high availability and high reliability to maintain a given system stable against sudden faults in the system. In this paper, we propose a new types of fault tolerant system based on a fault detection bus. The fault detection bus is designed and implemented to detect any errors by comparing event-output signals from two processor modules. It employs the hot standby sparing fault detection method[1] to provide continuity of services even if a system fault occurs. The prototype fault tolerant system is currently being implemented on a management system with two processor modules.

1. 서론

근래에 와서 시스템의 오동작이 치명적인 결과와 막대한 재산상의 문제를 초래할 수도 있는 실시간 시스템을 사용한 응용 분야가 점차 확산되고 있다. 특히 순간적으로 많은 양의 정보를 처리해야 하는 초고속 통신망에서의 제어 시스템은 고성능 및 고신뢰성이 기본적으로 요구되는 시스템으로서 이러한 통신망의 구조는 일반 범용 컴퓨터가 갖지 않는 실시간 연속성 서비스의 특성을 갖는다. 이러한 통신망의 제어시스템은 고장 발생에 따른 파급 효과가 클 수 있으므로, 고장 발생을 미연에 방지하거나 고장 발생시에 빠른 시간내에 고장을 검출하여, 정상적인 서비스를 계속하는데 문제가 발생하지 않도록 고장 전파를 막아야한다. 고장 발생을 미연에 방지하는 것은 한계가 있으므로 고장 발생시에 대처하는 방법에 대한 연구가 활발이

진행되고 있다.

시스템 하드웨어에 의해 발생할 수 있는 고장에 대해 고장 감내 시스템 실현을 위한 방법의 하나로 시스템 모듈들을 다중화 하여 서비스를 수행하고 있는 모듈에 문제가 발생시에 다른 모듈이 대신 서비스를 수행하도록 하고 있다. 다중화 방식에는 여러 방법들이 있는데 구현상의 어려움은 있지만, 많은 장점을 갖는 핫 스탠바이 스페어링 방식을 사용한 고장 감내 시스템 방식을 본 논문에서 다루고자 한다[1]. 핫 스탠바이 스페어링 방식은 다중화된 시스템 모듈들이 완벽하게 동기를 맞추어서 작업을 처리해야하고 시스템 모듈들의 레지스터나 메모리등의 시스템 자원들의 정보가 모두 같은 상태를 유지해야하므로 구현과, 운용에 많은 어려움이 있지만, 고장 발생시 절체 시간이 타 방식에 비해 극히 작아서 중단 없는 연속적인 서비스를 가능하게 하고, 에러가 전체 시스템에 확산될 수 있는 문제점을 제거 할 수 있어서 고가용성 및 고신뢰성을 요구하는 실시간 차세대 멀티미디어 서비스에 적합한 방식이다. 본 논문에서는 고성능 및 고신뢰성이 요구되는 초고속 정보 통신망에서의 제어 시스템에 적용 가능한 고장 검출 기능을 갖는 버스를 제안하고, 고장 감내 기능을 갖지 않는 상용 시스템에 I/O 작업시에만 고장을 검출함으로써 시스템 성능을 떨어뜨리지 않는 핫 스탠바이 스페어링 방식의 고장 감내 이중화 시스템을 설계 하였고 현재까지 구현된 내용을 소개하였다.

2. 고장 검출 버스 제안

고장 검출 버스는 각각 32비트 주소와 데이터 버스를 갖는 비동기 버스이다.

그림 1은 같은 하드웨어 컴포넌트를 갖는 Master 모듈과 Slave 모듈이 I/O Devices와 인터페이스 할 때 고장 검출 버스를 이용하는 것을 나타낸다. 각 모듈은 입출력 작업과 고장 검출을 위하여 고장 검출 버스 신호 생성부와 고장 검출부를 갖는다.

고장 검출 버스 신호 생성부는 IO Devices에 쓰기 및 읽기 작업을 할 때, 그에 적합한 신호를 생성 제공한다. 고장 검출 버스 신호 생성부에서 생성된 신호는 고장 검출 버스 기준 신호가 되고, Master에서 실제 출력된 고장 검출 버스 신호는 각 모듈의 고장 검출부로 입력되는 고장 검출 버스 비교 신호가 되어 기준 신호와 비교 작업을 하게 된다.

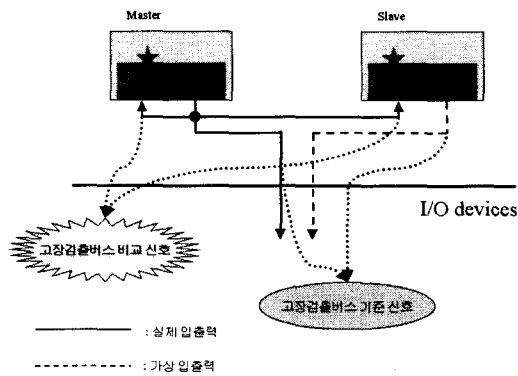


그림 1. 고장 검출 버스 신호

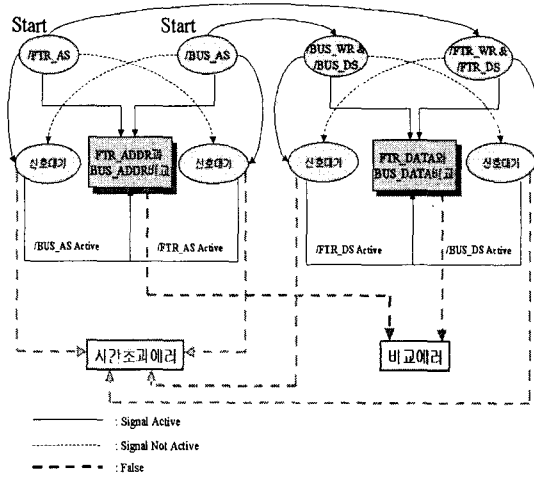


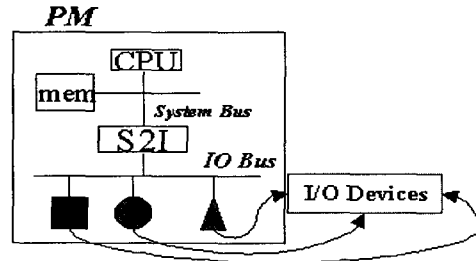
그림 2. 비교 및 시간초과 발생 상태 천이도

그림 2는 상태 천이도로서 이중화 모듈간 동기가 맞지 않아서 발생하는 시간초과 예러와 각 모듈이 I/O Devices에 제공하는 정보를 비교해서 서로 다를 경우 발생하는 비교 예러가 일어날 경우를 나타낸 그림이다. 그림에서 /FTR_* 신호들은 고장 검출 버스 신호

생성부에서 생성하는 신호이고, /BUS_* 신호들은 고장 검출부로 입력되는 신호를 가리킨다. 신호대기 상태에서는 일정시간동안 상대 신호가 Active 하지 않을 경우 시간초과 예러를 발생 시킨다. /BUS_AS와 /FTR_AS신호가 둘다 Active 일 때 서로의 주소를 비교하여 정보가 같지 않을 경우 비교 예러를 발생시키고, 데이터는 쓰기일 경우만 비교를 수행하는데, (/BUS_DS AND /BUS_WR) 이고 (/FTR_DS AND /FTR_WR) 일 때 서로의 데이터를 비교하여 같지 않을 경우 비교 예러를 발생시킨다. 그리고 읽기를 수행 할 경우는 읽는 데이터의 패리티를 검사하여 데이터의 오류를 검출 할 수 있도록 한다.

3. 고장 감내 이중화 시스템

3.1 고장 검출 버스 적용된 고장 감내 이중화 시스템 제안



S2I : System Bus를 IO Bus로 변환

그림 3. 단일 프로세서 모듈 시스템

그림 3은 고장 감내 방식이 적용되지 않은 일반 상용 시스템의 블록도이다. 다중화 방식을 사용하지 않고 단지 IO Bus에 I/O devices를 연결하여 입출력 작업을 수행한다. 이 방식은 입출력 동작시에 고장이 발생해도 시스템은 고장 발생 여부를 알지 못하게 되어있다. 그러나 그림 4에서 제안하는 고장 감내 이중화 시스템 하드웨어는 I/O devices를 제안된 고장 검출 버스(FT I/O BUS)를 통하여 연결하므로 입출력 작업시에 고장 발생 여부를 이중화 프로세서 모듈간 비교 작업을 통하여 알 수 있다. 그림 4는 본 논문에서 제안 할 고장 검출 버스를 적용한 고장 감내 이중화 시스템으로 핫 스탠바이 스페어링 기법을 사용한다. 따라서 직접 서비스를 수행하고 있는 MasterPM(Processor Module)에 고장 발생시에 가상적으로같은 서비스를 수행하고 있던 Standby PM이 MasterPM의 작업을 극히 짧은 절체 시간을 거쳐서 대신한다. 그래서 연속적인 서비스 수행을 가능하게

한다. FTMA와 FTIA는 고장을 검출 할 수 있도록 즉, 고장 검출 버스와 관련하여 인터페이스를 담당한다. FTMA는 고장 검출 버스 신호들을 생성하고, 고장을 검출하는 기능을 갖는다. FTIA는 고장 검출 버스를 이용하여 I/O Devices간 입출력 작업시 정보의 일시 저장 기능을 갖는다. Xbus는 이중화 프로세서 모듈간에 정보를 교환할 때 사용된다.

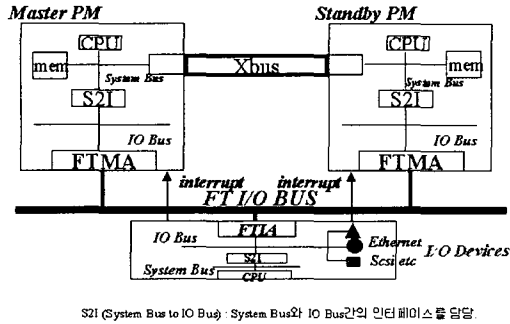


그림 4. 제안된 고장 감내형 이중화 시스템

3.2 핫 스탠바이 스페어링 고장 감내 이중화 시스템 운용 방법

그림 5는 고장 감내 이중화 시스템을 운용할 때 나타날 수 있는 프로세서 모듈들의 상태이다. 1)은 고장 감내 이중화 시스템이 서비스를 시작하기 전에 초기 상태로서 각 프로세서 모듈이 자체 동기 클럭으로 동작하며 이중화 모드로 전환하기 위한 준비 단계이다. 2)는 고장이 발생하지 않고 각 프로세서 모듈이 정상적으로 입출력 작업을 수행하며 동작하는 단계를 말한다. 이 단계에서는 두 프로세서 모듈이 밀접한 동기를 이루어서 같은 시각에 같은 작업을 하며, 같은 정보를 유지해야한다.

이때, Master 는 고장 검출 버스를 이용하여 실제 입출력을 수행하고, Slave는 가상적으로 작업을 한다. 3)은 2)번 그림과 같은 이중화 모드로 동작하고 있을 때 고장이 발생한 경우에 절체하는 것을 나타낸다. Master는 고장이 발생하여 자신은 자체 진단을 시작하고 같은 작업을 가상적으로 수행하던 Slave에 Master의 권한을 양도하여, 즉 Slave가 고장 검출 버스를 이용하여 입출력을 계속해서 수행할 수 있도록 한다. 4)는 고장이 발생했던 Master가 고장을 수리하고 난 뒤에 다시 이중화 모드로 동작하기 위한 준비 단계로 입출력 작업을 수행하고 있던 Slave의 변경된 자원을 Master에 넘겨주어서 각 프로세서 모듈이 같은 상태를 유지하게 한다.

4. 제안된 고장 감내 이중화 시스템에서 FTMA 및 FTIA 구현

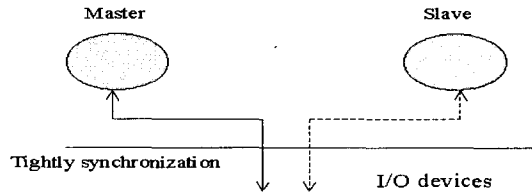
4.1 FTMA 블록 상세도

그림 6은 IO Bus라는 고장 검출 기능이 없는 상용 버스를 사용한 제어 시스템에 고장 검출 버스를 부가하기 위해 만들어진 FTMA 블록도이다.

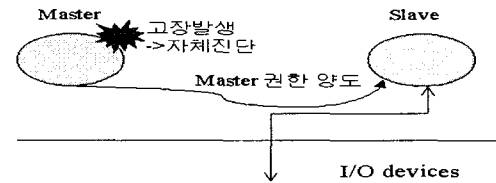
1) 독립동작 (초기 모드)



2) 정상동작 (이중화 모드)



3) 고장발생 (분리 모드)



4) 고장수리 (재설정 모드)

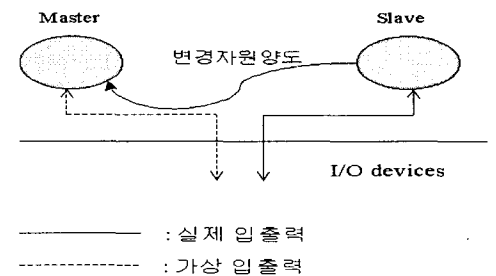


그림 5. 고장 감내 이중화 시스템 운용 모드

FTMA는 그림 4에서 제안하였듯이 이중화 모듈 프로세서 보드에 고장 검출 버스(FTIO BUS)를 생성하고 제공하기 위해 만든 Daughter 보드이다. FTMA는

