

여분을 갖는 통신시스템에서의 자기검사특성 향상에 관한 연구

A Study on Enhanced Self-Checking Characteristics in Redundant Communication System

신덕호* 김현기* 이재훈* 이창훈** 이기서***

Shin Duck Ho, Kim Hyun Ki, Lee Jea Hoon, Lee Chang Hoon, Lee Key Seo

ABSTRACT

In this paper, the method is used that each system isolates the part in which faults are occurred in case of fault occurrence since total system is designed in the type of hot standby sparing system construction and self-checking circuit in each system is attached to the memory being of the high failure rate in common communication system. Thus reliability and availability in system are improved.

In this thesis, self-checking circuit was designed using 2 out of 6 code and didn't have any effect to the system operation.

For the purpose of verification, each system attached with self-checking circuit was simulated through fault injection and the continuity of each system in hot standby sparing system was tested in case that the system was switched when faults in the system were occurred.

1. 서 론

여분을 갖는 시스템의 출현은 시스템의 오동작을 방지하기 위한 기본적인 대책이었다. 현재 대기중계, 다수결 시스템 등 여러 형태의 시스템이 개발되었으나 시스템의 고성능화와 통신기술의 발전에 의한 시스템 분산화에 따라 기존의 시스템을 보다 효율적으로 관리하기 위한 연구가 필요하게 되었다.

여분을 갖는 시스템을 구성하기 위해, 필요한 결합의 검출 회로를 자기-검사 회로로 설계하였고 결합 검출 회로가 장치된 시스템을 통신으로 연결함으로써 이중계 시스템으로 구성하였다.

시스템의 결합을 검출하기 위한 방법으로는 크게 On line과 Off line 테스트가 있으며, 본 논문에서는 On line 테스트의 가장 대표적인 자기-검사 회로(Self-Checking)를 설계하였다.

본 논문에서 설계된 자기-검사 회로는 통신 데이터가 쓰여질 메모리를 사전에 검증하는 역할을 하기 위해 이러한 동작을 하는 자기-검사 회로를 m-out-of-n checker로써 설계하였으며, 규약화 되어있는 코드를 각각의 단일 시스템의 통신버퍼에 병렬로 적용하여 데이터가 저장될 메모리 영역의 결합 발생 여부를 체크한다.

* : 광운대학교 제어계측공학과 박사과정

** : 광운대학교 제어계측공학과 석사과정

*** : 광운대학교 제어계측공학과 교수

또한 본 논문에서 제시된 자기-검사 회로를 통신 보드에 적용해서 설계함으로써 전용의 시스템에서 사용할 수 있는 시스템을 구현하였고, 이러한 시스템을 여러 개 두어 여분을 갖는 시스템으로 구성하여 가용성을 향상시켰다.

2. 본 문

2.1. 결함의 분석

본 논문에서는 통신을 이용한 제어 시스템을 결함허용 시스템으로 재구성하기 위해 구체적인 방법과 설계를 제시하였다. 설계를 통한 연구의 목적은 크게 3가지로 나뉜다.

- 자기-검사 회로를 추가하여 통신용 버퍼의 결함검출
- 하드웨어 여분을 이용한 시스템의 가용도 향상
- 신뢰할 수 있는 데이터를 통한 하드웨어 여분의 관리

CPU보드의 핵심 부품들의 신뢰도를 상용소프트웨어(Relax)로 계산한 결과가 그림 1과 같이 나타났다.

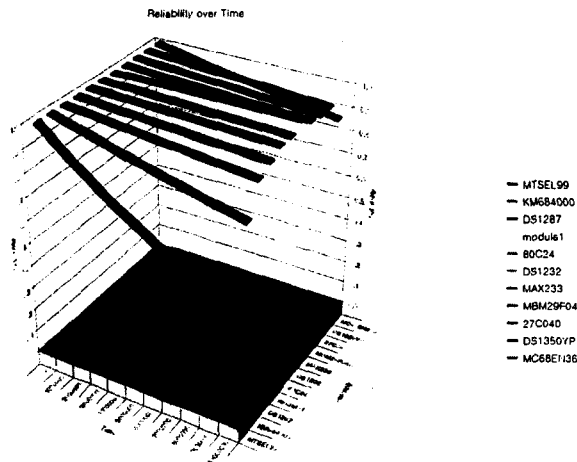


그림 1. 부품별 신뢰도

그림 1에서 볼 수 있는 것처럼 메모리의 신뢰도가 가장 떨어짐을 알 수 있다. 통신시스템과 같이 메모리의 신뢰성에 시스템의 신뢰도가 종속되는 시스템의 메모리신뢰성 향상을 목적으로 논문의 핵심인 자기-검사 회로를 설계하게 되었다.

본 연구에 의해 설계된 CPU 보드는 상용의 실시간운영체제가 동작할 수 있는 여건을 갖추어 신뢰성을 향상시켰으며, 대기 이중계 시스템을 구성하여 실험을 수행하고 임의의 결함을 발생시켜 가용성을 증명하였다.

2.2 자기-검사 회로

자기-검사란 외부의 테스트 장치가 없어도 시스템 내부의 어떤 결함이라도 자동적으로 검사해 낼 수 있는 능력을 말하며, 자기-검사 회로의 설계는 체커라고 불리는 서브회로에 의한 자동적인 결함 검출을 할 수 있는 회로를 말한다. 이 회로에는 여러가지 코드가 사용되는데 코딩 이론의 개념이 바탕이 되어야 한다. 특정 코드의 종류에는 parity-check codes, beger codes and residue code 등이 있다.

자기-검사 논리의 개념을 형식화하기 위해서는 결함-안전(fault-secure), 자기-시험(self-testing) 그리고 전체 자기-검사(totally self-checking)를 정의해야 한다.

1) 결함-안전(Fault-Secure)

결함이 출력에 아무 영향을 주지 않거나 출력에 영향을 미쳐도 출력된 값이 유효하지 않은 code word면 이 회로를 fault secure하다고 한다. 반대로 결함이 발생하여 출력된 값이 틀린 code word 이지만 유효한 code word 이면, 이 회로는 fault secure라고 하지 않는다.

임의의 회로에서 어떤 단일 결함이 존재한다면 그 회로는 어떤 정확한 입력 코드워드에 대해서 (N) 정확한 코드워드를 생산하지도 ($\notin S$), noncode word를 생산하지도 않는 것을 결함-안전이라고 말한다. $F, Y(X, f) = Y(X, \lambda)$ (null fault) 또는 $Y(X, f) \notin S$ 는 fault secure(FS)이다. 다시 말하면, 회로 상에서의 결함이 출력에 어떠한 영향도 주지 않고, 효과도 없는 것을 결함-안전이라고 한다.

2) 자기-시험(Self-Testing)

각각의 입력 가능한 code word가 결함은 발생되지 않고 유효한 출력 code word를 발생하는 단일 결함도 검출 가능한 회로를 self-testing이라 한다.

3) 전체 자기-검사(Totally Self-Checking)

fault secure와 self-testing을 만족하면 totally self-checking이라고 한다. totally self-checking은 적어도 하나의 유효한 입력 code word에 의해서 결함을 검지 할 수 있고, 입력의 조합이 결함을 검지 하지 못하여도 출력은 정확한 code word를 발생하는 회로에 해당된다.

자기-검사에 대한 이론에 대한 가정은 아래와 같다.

가정1. 검사기를 통해 발생하는 모든 결함들은 단일결함에 해당된다.

가정2. 두 개의 결함이 발생하는 사이에 충분한 시간지연(Time Delay)이 필요하며 시스템은 결함 검출 기술에 의해서 충분히 에러를 검출 할 수 있다.

위의 가정에 의한 함수 블록과 검사기들은 다음과 같이 정의 될 수 있다.

정의 1.

전체 시스템 G에서 단일 입력이 임의의 올바른 입력 코드 공간에서 correct code space 또는 non correct code space를 출력할 때 결함-안전이라고 한다. 결함이 출력에 영향을 받지 않거나 영향을 받더라도 non output code space 일 때 결함-안전이라고 한다.

정의 2.

회로상에 단일 결함이 존재할 때 적어도 하나의 정확한 입력 코드 워드가 존재하고 그 입력 코드에 대해서 정확하지 않은 코드 워드를 출력 한다면 자기-시험이라고 한다. 다시 말하면, 결함 f가 검출되는 동안 적어도 한번은 testing mode상에 결함이 존재 할 수 있다.

정의 3.

전체 시스템 G에서 코드-분리(code-disjoint)는 입력 X의 원소 x와 x'이 다음과 같은 논리 일 때

코드-분리라고 한다.

$$\forall x \in X, G(x, \emptyset) \in Y, \text{ and } \forall x \in X, G(x, \emptyset) \notin Y$$

정의 4.

위 정의 1,2,3을 만족 할 때 완전한 자기-검사(Totally Self-Checking)라고 말한다.

2.3 자기-검사 특성

self-checking logic 개념은 전형적인 “checking the checker”문제로부터 시작된다. coding scheme 또는 duplication with comparison이 설계에 많이 이용되지만, 두개의 모듈의 출력을 비교할 때 또는 출력된 값이 유효한 code word 인지를 평가하기 위해 필요할 뿐이다. 이러한 기술의 근본적인 문제는 비교기 또는 code 체커의 정확한 동작을 위한 신뢰성 있는 접근이다. 만약 code 체커가 고장났을 경우 예를 들면, 에러가 발생하지 않았는데도 시스템이 에러가 발생했다고 하거나 또는 에러가 발생해도 시스템이 적합한 에러의 검출을 하지 못하는 경우이다.

어플리케이션을 수행하는 많은 구조에서 이러한 상태는 수용될 수 없다. 비교기와 code 체커의 설계 시에 이들이 자신의 결함을 검출할 수 있는 설계가 제공되어야 한다.

일반적으로 어떤 회로가 외부적인 자극에 의해서가 아닌 자동적으로 존재하는 결함을 검출할 때에 이 회로를 self-checking이라 한다. 다시 말하면, self-checking 회로는 정상동작에서의 결함의 존재를 판단할 수 있어야 한다. self-checking logic은 일반적으로 coding 기술을 사용하여 설계된다.

회로설계의 기본개념은 결함이 없는 상태에서 유효한 값이 입력되면, 정확한 출력 code word를 내보내고, 만약 결함이 존재해도 유효하지 않은 code word를 출력하여 결함이 검출되었음을 알리는 것이다.

self-checking logic 개념의 형식을 갖추어 fault secure, self-testing 그리고 totally self-checking을 정의하였다. 이 각각이 정의는 입력 라인에서의 유효한 code word와 출력 라인에서의 code word가 발생하는 회로의 설계 시에 인지해야 하는 것들이다.

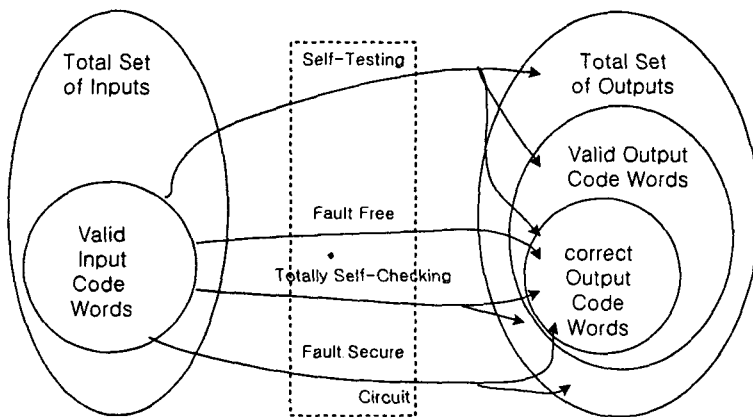


그림 2. Self-testing fault secure totally self checking의 개념도

그림 2는 totally self checking circuit의 특성에 대한 정의를 설명한다. 초기의 입력 값은 모든

입력들의 조합에 일 부분인 유효한 입력 code word로 encode된다. 예를 들어, 4-bit binary number의 16개 가능한 수들 중에서 odd-parity code words는 8개만이 만족하는 조건을 갖는다. 유사하게 모든 출력 code word는 유효한 출력 code word로 분할되고, 더 나아가 정확한 출력 code word로 분할된다. 정상동작에서 결함이 없는 회로는 유효한 입력(valid input code word)을 받아 정확한(correct) 출력 code word를 생산한다. fault secure 회로의 경우 유효한 입력 code word를 허용하여 결함이 존재할 경우 정확한 출력 code word 또는 noncode word를 내보낸다.

2.4 2 out of 6 Self-Checking Logic 설계

2-out-of-6 Self-checking Checker의 설계는 아래와 같은 함수에 근거한다.

$$2\text{-out-of-6 code } A = \{x_0, x_1, x_2\} \quad B = \{x_3, x_4, x_5\}$$

$m=2, n=6$, 임의로 출력함수를 구성하면 출력함수 Z_0, Z_1 은 다음과 같다.

$$\begin{aligned} f_0 &= \sum_{i=0}^1 T(m_A \geq i) \cdot T(m_B \geq 2-i) \quad i: \text{even} \\ &= (m_A \geq 1) \cdot (m_B \geq 2-1) \\ &= (x_0 + x_1 + x_2) \cdot (x_3 + x_4 + x_5) \end{aligned}$$

$$\begin{aligned} f_1 &= \sum_{i=0}^1 T(m_A \geq i) \cdot T(m_B \geq 2-i) \quad i: \text{even} \\ &= 0 \quad A_1 = \{x_0\}, A_2 = \{x_1, x_2\}, n_{A1} = 1, n_{A2} = 2 \end{aligned}$$

$$\begin{aligned} f_2 &= \sum_{i=0}^1 T(m_{A_1} \geq i) \cdot T(m_{A_2} \geq 2-i) \quad i: \text{odd} \\ &= x_0(x_1 + x_2) \end{aligned}$$

$$\begin{aligned} f_3 &= \sum_{i=0}^1 T(m_A \geq i) \cdot T(m_{B_1} \geq 2-i) \quad i: \text{even} \\ &B_1 = \{x_3\}, B_2 = \{x_4, x_5\}, n_{B1} = 1, n_{B2} = 2 \end{aligned}$$

$$\begin{aligned} f_4 &= \sum_{i=0}^1 T(m_B \geq i) \cdot T(m_{B_2} \geq 2-i) \quad i: \text{odd} \\ &= x_3(x_4 + x_5) \end{aligned}$$

$$\begin{aligned} f_5 &= \sum_{i=0}^1 T(m_{B_1} \geq i) \cdot T(m_{B_2} \geq 2-i) \quad i: \text{even} \\ &= x_4x_5 \end{aligned}$$

$$z_0 = (f_0 \cdot f_2 \cdot f_4)(f_0 \cdot f_5) + (f_2 \cdot f_4)(f_3 \cdot f_5)$$

$$z_1 = (f_0 \cdot f_2 \cdot f_4 + f_0 \cdot f_5) \cdot (f_3 \cdot f_5 + f_2 \cdot f_4)$$

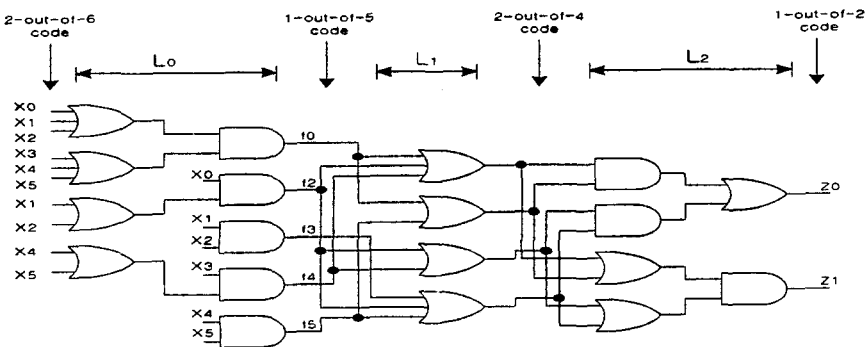


그림 3. 2-out-of-6 checker

2.5 목표시스템

통신 기반의 시스템은 데이터 통신을 이용하여 제어를 하게 된다.

본 논문에서는 MC68EN360 통신 프로세서를 이용하여 시리얼 통신 및 이더넷 통신을 제어하는 시스템을 이중계로 구성하였다.

MC68EN360은 CPU32+라는 CPU 코어를 사용하며, 통신 부분은 4개의 SMC(Serial Management Controller)를 이용하여 시리얼과 이더넷 통신을 한다.

각각의 SMC는 통신 버퍼로 설정된 메모리에 입력된 모든 데이터를 기록하며, 이더넷의 경우 모든 패킷을 버퍼에 저장한 후에 자신의 수신 아이디가 들어있는 패킷을 검색하게 되어있다.

시스템의 모든 정보와 제어명령이 이 통신 버퍼에 저장되므로 통신 버퍼에 자기-검사 회로를 설치하여, 데이터가 저장될 공간의 신뢰성을 확보하는 것이 목표이다.

소프트웨어의 구성은 실시간 운영체제를 이용하여 어플리케이션과 자기-검사 회로를 위한 타스크를 분리하여 업로드시켜 범용성과 확장성이 용이하게 설계하였다.

2.6 Self-Checking Logic 설계의 적용

자기-검사 회로를 설계하여 그림 4와 같이 시스템에 설치하였다.

그림 12에서 자기-검사 회로는 컨트롤러에서 나오는 유효한 코드를 메모리에서부터 읽어들이며 메모리의 안전성을 확보한다.

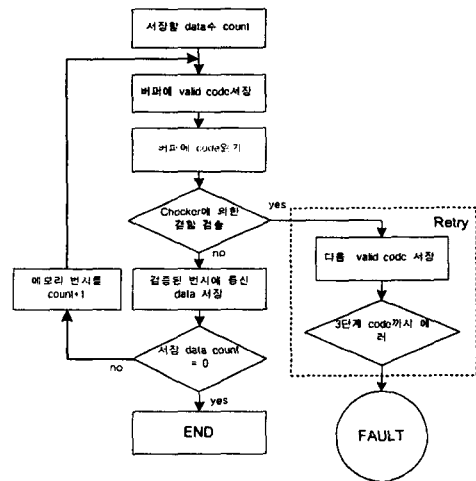
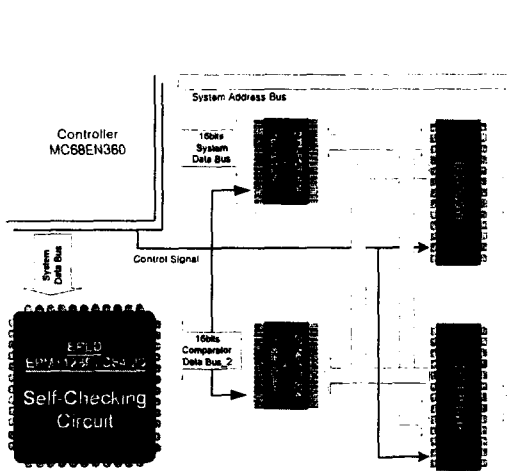


그림 4. 자기-검사 회로의 설치 그림 5. 자기-검사회로 환경을 위한 소프트웨어 순서도.

컨트롤러는 메모리에 유효한 코드를 저장한다. 이때의 데이터 버스는 컨트롤러에서 메모리 쪽으로 버퍼에 의해 연결되어, 어드레스 버스가 가리키는 번지에 데이터를 저장하게 된다. 컨트롤러가 유효한 코드를 검증이 필요한 번지에 저장한 후에 컨트롤러 리드 싸이클이 들어오면 자기-검사 회로를 인에이블 시킨다, 자기검사 회로는 메모리를 리드 이네이블 시켜 자기-검사 회로로 데이터를 읽어 들인다. 메모리의 데이터를 읽어 들일 때는 16bits 버퍼에 의해 시스템 데이터 버스와 메모리의 데이터 버스를 분리시켜, 컨트롤러의 동작에 관계없이 자기-검사 회로가 데이터를 체크할 수 있는 환경을 만들어 간다.

자기-검사 회로의 내부의 블록도는 그림 6과 같이 설계되어, 하드웨어적으로 코드를 체크하게

된다.

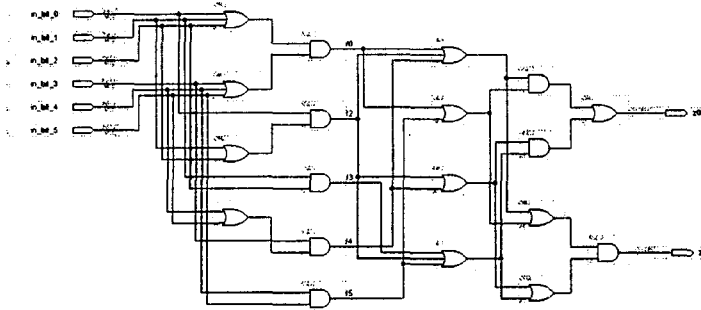


그림 6. 자기-검사 회로의 내부

자기-검사 회로의 구동을 위한 컨트롤러의 소프트웨어 플로어차트는 그림 5와 같이 설계되어 독립된 태스크로 운영체제에 의해 제어된다.

3. 결 론

여분을 이용한 결함허용 시스템을 구성함에 있어, CPU 보드 각각의 통신 버퍼에 자기-진단 로직을 설계하고, 결함을 검출하여 신뢰할 수 있는 통신 버퍼를 통한 여분을 갖는 통신시스템을 구성하였다.

이러한 구조는 일반적인 통신 시스템에서 사용되는 일반적인 통신보드에 비하여 통신전용 메모리에서 발생하는 결함에 대한 검출이 가능하여, 보다 신뢰성 있는 데이터 통신을 가능하게 하였다.

자기-진단 로직에는 2-out-of-6 체커를 사용하여 EPLD로 구현하였으며, 실제의 보드에 자기-진단 로직을 구성하여 메모리의 결함을 검출하였다.

실험에서는 통신 메모리 부에 강제적인 결함을 발생시켜 결함이 검출된 보드를 효과적으로 절체 시킴으로써 결함을 제한하는데 성공하였다.

이러한 구조의 여분을 갖는 시스템은 높은 신뢰성과 가용성을 요구하는 통신시스템에 유용하게 사용되리라 생각된다.

향후에는 상용 컨트롤러의 가용성 향상과 통신메모리의 신뢰성 보장 외에도 시스템에 사용되는 다른 디바이스들에 대한 결함허용 구조를 구성하는데 역점을 두어야 할 것이다.

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.

- [4] "Optimal Self-Testing Embedded Parity Checkers" Dmitris Nikolos, Member, IEEE Vol. 47. No. 3. MARCH 1998.
- [5] "Error Secure/Propagating Concept and its Application to the Design of Strongly fault-Secure Processors" TAKASHI NANYA, Member, IEEE and TOSHIAKI KAWAMURA. Vol.37. No.1. JAN. 1988
- [6] "On-Line Detection of Bridging and Delay Faults in Functional Blocks of CMOS Self-Checking Circuits" Cecilia Metra, Michele Favalli, Piero Olivo, and Bruno IEEE. Computer-Aided Design JULY 1997 Vol 16. No. 7.
- [7] "Design of Totally Self-Checking Circuits for m-out-of-n Codes" IEEE Trans. Computers. Vol. 22.No. 3. pp. 263-169. Mar. 1973
- [8] "SAFETY-CRITICAL COMPUTER SYSTEMS" Written by Neil Storey 1996