

고장-안전 제어논리의 신뢰성 특성에 관한 연구

한 상 섭* 김 민 수** 이 정 석*** 이 기 서***

* 삼성전자 CS 경영센터 전문기술 그룹
** 동해대학교 정보통신공학과
*** 광운대학교 정보제어공학과

A Study On The Reliability Characteristics of Fail-Safe Control Logic

Sang Seop Han*, Min Su Kim**, Jung Suk Lee***, Kee Seo Lee***

* SEC CS Center Advanced Technology Group Engineer
** Information & Communication Engineering Department, Dong-Hae Univ.
*** Information & Control Engineering Department, Kwang-woon Univ.

Abstract

This paper modelled the fail-safe control logic through the frequency coding input and designed the 3-out of-6 self checker using the error detect coding method of information redundancy. In addition, this paper also performed the reliability parallel numeric analysis regarding single module between fail-safe control logic module and TMR(Triple Modular Redundancy), therefore, we achieved the result that the fail-safe control logic increases a functional reliability because of decreasing system waste cost and functional overhead rather than the existing hardware redundancy method.

1. 서론

임의의 시스템이 어느 정도의 신뢰성을 갖고 동작하며 어떤 결함이 발생 가능한지는 공학을 하는 사람들로써는 한 번쯤 관심을 갖지 않을 수 없다. 본 논문은 보다 나은 신뢰성 구현을 위해 고장-안전 제어 논리[3]를 모델링 하고 기존의 신뢰성을 위한(TMR) 모델[1]과 어떤 점이 다른지 비교한다. 임계-안전 시스템[8] 구성에 있어서 집적 회로의 사용은 점점 더 중요하게 생각되고 있으며 병행 에러 검출과 자체검사 회로 구성을 요구하게 된다. 자체 검사 회로의 일반적인 구성은 기능 블록의 출력과 함께 검사기로 구성된다.[9] 이 기능 블록

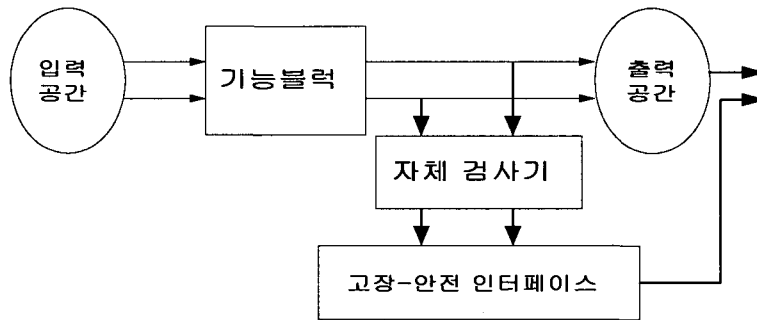
의 출력들은 암호화[9]되어 있으며 검사기들은 그 출력 값이 코드 공간에 속해 있는지를 증명하기 위해서 사용된다. 기능 블록이 주어진 코드공간 외의 출력을 발생하면 검사기는 에러를 가리킨다. 자체 검사기는 다양한 검사기에서 출력된 다양한 에러들을 지적하게 되며 이러한 자체 검사회로와 고장-안전 인터페이스의 구성으로 고장-안전 제어 시스템을 모델링 할 수 있다.[13]고-안전(High-Safety) 시스템의 설계를 위해서 시스템 전체로부터 보드레벨에 이르기까지 단위 모듈별로 자체 검사능력을 갖추어야 한다.[3] 신뢰성 향상을 위한 하드웨어 여분은 안정성을 높이는 대신 높은 비용과 가용성이 감소되는 단점 [1][2][3]이 있다. 그러나, 고장-안전 제어 시스템은 안정성을 높이면서 하드웨어 중복의 단점을 보완 할 수 있다는 것을 신뢰성 비교 분석으로 알 수 있다.

2. 본론

2.1 자체-검사 와 고장-안전 제어 논리의 정의

고장-안전 제어 논리 설계를 위한 자체 검사회로의 일반적인 구조는 기능 블록의 출력과 검사기의 관계로 정의[1][2][9]한다.

2.1.1 정의[3][7]



<그림 1> 고장 안전 제어 논리 블록도

디지털 논리 회로의 다중 입, 출력에 대한 자체 검사 및 고장 안전 기능은 회로 정상 동작 동안 모든 입력 벡터 집합에 대해서 입력 공간의 부 집합(Subset)을 포함하고 이 부 집합(Subset)은 입력 코드 워드(Input Code Word)라고 부른다. 입력 벡터 중 올바르지 않은 공간을 입력 비-코드워드(Input Non-Code Words)라고 한다. 또한 출력 방정식 또는 함수 식에 따른 모든 출력에 대해서 출력 벡터 집합이라고 하며 이 출력 벡터의 부 집합(Subset)을 출력 공간이라고 한다. 이러한 출력 공간 중 출력 방정식 및 함수 식에 대해서 올바르지 않은 출력 집합을 출력 비-코드워드(Output Non-Code words)라고 한다. 이러한 기본적인 속성을 다음과 같은 디지털 회로의 자체-검사 및 고장-안전 속성을 정의 할 수 있다. 디지털 논리 회로가 결함 집합상의 모든 결함에 대해서 모든 입력 코드 워드들이 올바르지 않는 출

력 코드 워드들을 결코 생산하지 않는다면 그 회로는 결함 집합에 대한 결함 안전 (Fail-Safe(FS))이고 디지털 논리회로가 결함 집합상의 모든 결함에 대해서 입력 코드 워드 중 적어도 하나가 올바르지 못한 출력 코드워드를 생산한다면 결함 집합은 그 회로에 대한 자체 시험(Self-Testing(ST)) 이다. 또한 디지털 논리 회로가 결함 집합에 대해 FS(Fail-Safe) 그리고 ST(Self-Testing)를 만족한다면 그 회로는 TSC(Totally Self-Checking)이다. 이러한 자체 검사 특성에 다음과 같은 코드분리정의를 만족시키면 고장 안전 제어 특성을 구성 할 수 있다. 디지털 논리 회로가 올바른 출력 코드 워드를 통해 올바른 출력 코드 공간이 사상되거나 올바르지 못한 출력 코드 공간을 통해 올바르지 않는 입력 공간으로 항상 사상된다면 코드 분리(Code-Disjoint(CD))라고 한다.

2.2 자체 검사기 설계와 고장 안전 인터페이스의 설계

위와 같은 정의에 따른 3-out-of-6 코드검사기의 설계는 다음과 같다. 정보여분기법을 이용한 m-out-of-n 코드의 특성은 올바른 코드워드 공간에서 m 개의 “1”의 개수와 (n - m) 의 “0”의 개수를 갖으며 m-out-of-n 코드 검사기는 두 개의 독립 서브회로들이 단일 출력 값을 갖도록 구성된다. 올바르지 않은 코드워드 입력에 대해서 검사기의 “1”의 개수가 m보다 많거나 적게되면 검사기의 출력은 (1,1) 또는 (0,0)이 되며 정상적인 출력일 경우 (1,0) 또는 (0,1)이 된다. 이 때 출력 값은 다음과 같이 구할 수 있다.

$$z_0 = \sum_{i=0}^m T(m_A \geq i) \cdot T(m_B \geq m - i) : i = \text{odd integer} \quad (1)$$

$$z_1 = \sum_{i=0}^m T(m_A \geq i) \cdot T(m_B \geq m - i) : i = \text{even integer} \quad (2)$$

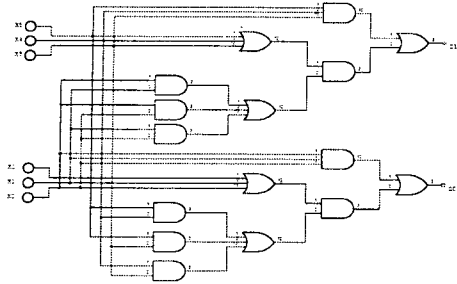
(단, m_A 와 m_B 는 서브회로 집합 A, B의 원소이다.)

출력 방정식에 따른 3-out-of-6 코드 검사기는 (n = 6 , m = 3)이므로 두 서브회로를 구성하는 A = (x_0, x_1, x_2)와 B = (x_3, x_4, x_5)로 초기화를 하고 다음과 같은 출력 값이 유도된다.

$$\begin{aligned} z_0 &= T(m_A \geq 1) \cdot T(m_B \geq 2) + T(m_A \geq 3) \cdot T(m_B \geq 0) \\ &= (x_0 + x_1 + x_2)(x_3 x_4 + x_4 x_5 + x_5 x_3) + x_1 x_2 x_3 \cdot 1 \end{aligned} \quad (3)$$

$$\begin{aligned} z_1 &= T(m_A \geq 0) \cdot T(m_B \geq 3) + T(m_A \geq 2) \cdot T(m_B \geq 1) \\ &= 1 \cdot x_3 x_4 x_5 + (x_0 x_1 + x_1 x_2 + x_0 x_2)(x_3 + x_4 + x_5) \end{aligned} \quad (4)$$

위와 같이 설계된 자체-검사기에 주기적인 코드를 입력하여(하드웨어적으로) 단일 결함 검출의 단점을 보완 할 수 있다. 또한 회로 집적화를 위해 고장 안전 인터페이스 자체를 자체 시험 가능하도록 설계(FPGA)할 수도 있다. 본 논문에서는 실시간 기반 결함 허용 시스템에



<그림 2> 3-out-of-6 코드 자체 검사기

<표 1> 3-out-of-6 코드 검사기 출력

Sub Set 1 (Z0)			Sub Set 2 (Z1)		
x0	x1	x2	x3	x4	x5
0	0	0	1	1	1
1	0	0	0	1	1
0	1	0	1	0	1
0	0	1	1	1	0
1	1	0	0	0	1
0	1	1	1	0	0
1	0	1	0	1	0
1	1	1	0	0	0

서 하드웨어 여분을 이용한 TMR 구조와 동적 여분 기법인 고장 안전 구조에 대해서 정형 (형식) 기법을 이용 주기적인 코드를 주입하여 소프트웨어적 실험을 하였다.

2.3 신뢰성 해석과 정형 기법(Formal Methods)

정형 기법 검증 툴인 SMV(Symbolic Model Verification)를 이용하여 자체 검사기의 조합 회로에 대한 주기적인 코드를 주입하고 3-out-of-6 고장 안전 인터페이스 제어 논리에 대한 검증을 했다.¹⁾ 주기적인 코드 주입의 의미는 정해진 코드영역만을 검사하는 자체 검사기의 단점을 보완하고 다중 결함 검출 문제를 위한 것이다. 즉, 주기적인 코드 주입을 통해 분리된 코드 영역에 대한 연속적인 코드영역의 검사가 이루어질 수 있다. 자체-검사-기반 결함 허용 시스템과 기존의 TMR 구조는 안정성과 하드웨어적 비용에 있어서의 비교를 할 수 있다. 이 신뢰성 해석은 모듈 단위의 평가로써 임의의 기능논리에 해당한다. TMR 보터 (Voter) 기술은 주어진 임무 완료 시간 동안 신뢰성 R 을 갖는다면[1], 각각의 모듈은 주어진 시간 T 에서 신뢰성 R^T 로 각각의 모듈을 나타낼 수 있다. 보터가 고장났거나 다른 두 모듈이 고장나고 나머지 하나의 모듈만이 정상동작일 경우 TMR 시스템은 고장이다.[1] 그러나, 보터(보터 신뢰성을 1로 가정)와 두 개의 모듈이 정상이면 정확한 결과를 산출 할 수 있다.

R_v : 투표기의 신뢰도, R_m : 모듈의 신뢰도, R_s : 전체 신뢰도

$$R_s = R_v R_m^3 = R_v (R_m^3 + 3R_m^2(1 - R_m))$$

5) Formal Methods : Formal Methods는 H/W 설계 에러들을 제거하기 위한 수학적 모델 에(이산 수학) 의존하는 분석적 접근이며 궤환(Feedback) 또는 비 궤환(Non-Feedback) 루프를 갖는 제어 시스템은 수학적으로 표현이 되며 이러한 수학적 표현이 정형 기법의 표현과 상통하고 그 수학적 표현은 이산 수학으로 표현하고 있다. 본 논문에서는 SMV 툴을 이용하여 고장 안전 제어 논리에 대한 검증을 실험했으며 상용 버전의 소프트웨어가 없는 관계로 아래 자료를 참고 바람. (Linux Version : 개발자 툴) 또한 PVS(NASA), VDM, Z, Murphi 등과 같이 기계 역학이나 항공 및 신호 논리에 대한 검증을 위한 소프트웨어들이 세계적으로 연구되고 있으며 개발 단계에 있다.
참고자료 : <http://www.microtrack.co.kr/자료실/RAMS/SMV/>

$$R_M = (3\text{모듈이 모두 정상일 확률}) + (2\text{ 모듈만 정상일 확률})$$

$$= R^3_m + \binom{3}{2} R^2_m(1 - R_m) \quad (5)$$

TMR 시스템이 한 개의 모듈로 구성된 시스템 보다 신뢰도가 높기 위해서는 즉, $R_{System} > R_{Module}$ 만족해야 하며 $R_M > 0.5$ 이어야 한다.[2][3][4][5] 여기서 TMR 시스템 안전성은 S_{TMR} 이라고 하면 신뢰성과 안전성의 관계는 $S_{TMR} = R$ 이라고 할 수 있다. 자체-검사 모듈 상에서의 신뢰성은 δ (δ :Self-Checking Checker의 생명주기(Life Cycle Time))시간 동안의 신뢰성으로 표현 ($R^{\delta T}$, T: TMR 시스템 생명주기(Life Cycle Time))하면 신뢰도 $R = 0.99$ 와 $R = 0.999$ 에 가까이 근접하는 경우 우리는 각 시스템의 신뢰성을 (6), (7)에서 기대할 수 있고 병렬 시스템 신뢰성을 상대 비교하여 두 시스템의 신뢰성을 비교 할 수 있다. 이 두 시스템의 비교는 시스템의 하드웨어적 비용 감소화 문제와 신뢰성 측면에서 하드웨어의 여분 기법보다는 임계 안정 시스템의 구축이 우수하다는 것을 알 수 있다. TMR과 고장 안전 시스템의 신뢰성을 알아보면

$$R_{TMR} = [R^T \cdot R^T \cdot R^T + 3 \cdot R^T \cdot R^T(1 - R^T)] \cdot R = (3R^{2T} - 2R^{3T}) \cdot R \quad (6)$$

$$R_{SC} = [R^{\delta T} \cdot R^{\delta T} + 2R^{\delta T} \cdot (1 - R^{\delta T})] \cdot R = (2R^{\delta T} - R^{2\delta T}) \cdot R \quad (7)$$

병렬 해석을 하면 $R(t) + Q(t) = 1.0$ 에서

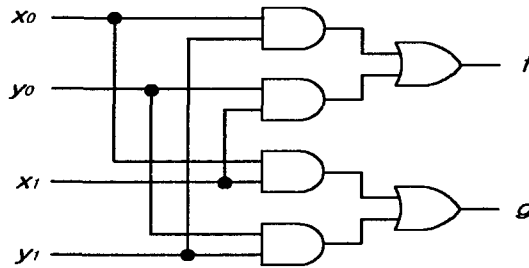
$R_p(t) = 1.0 - Q_p(t) = 1.0 - \prod_{i=1}^N Q_i(t) = 1.0 - \prod_{i=1}^N (1.0 - R_i(t))$ (N :고장개수)
 이 모듈이 2개일 경우 : $1 - (1 - R_c(t))^2$ 이다. (R_c : 자체 검사기의 신뢰도) $0 \leq R^{\delta T} \leq 1$ 에서 $\delta \rightarrow \text{limit}$, $R^T \rightarrow 1$ 이라면 즉, 임의의 시간에서 자체 검사기가 고장일 경우 그때의 TMR 시스템 신뢰성을 1로 가정하면, 다음과 같은 신뢰성 비교식을 얻을 수 있다.

$$\lim_{R^T \rightarrow 1} \frac{\ln\{1 - \sqrt{[1 - 3(R^T)^2 + 2(R^T)^3]}\}}{\ln(R^T)} = \sqrt{3} = 1.732 \quad (8)$$

(8)의 의미는 자체 검사기 모듈이 이상적인 에러 검출 코딩과 정확한 자체 검사 정보 여분 기법의 검사기일 경우 일반적으로 Two-rail 형태를 갖게 되므로(<표 1> 참조) 2개의 모듈 신뢰성 함수식에 TMR의 신뢰성 함수를 (8)과 같이 대입하면 TMR 시스템에 비해서 자체-검사 모듈의 오버헤드가 73%를 초과하지 ($[\sqrt{3} - 1]100\%$) 낮는다는 것을 알 수 있다. Two-rail 따라서, 고장-안전 제어 논리의 기능 모듈 신뢰성이 TMR 구조의 결함 허용 기능 모듈보다 높은 기능 신뢰성(시스템에 미치는 오버헤드)특성을 갖는다고 할 수 있다.

Two-rail 검사기는 입력에 대해서는 x_n, y_n 의 두 그룹을 가지고 있으며, 출력에 대해서는 f, g 의 두 그룹을 가지고 있다. 출력 측에서 발견된 신호는 항상 보완되어야 하며, 1-out of-2 code의 형태를 띄고 있으며 x_j, y_j 가 또한 모든 j (이때, $1 \leq j \leq n$)에 대해서 만족하는 것과 같은 필요 충분조건을 만족한다. 이러한 방법에 대해서는, 출력 $y_j = \overline{x_j}$ 값을 갖는 아래

그림을 참고하면 쉽게 이해할 수 있을 것이다. $x_0x_1=11, y_0y_1=00$ 값을 가질 때, 예러가 없는 상황에서는 이 입력 값에 따른 결과 값 $f=0, g=1$ 을 갖는다. 결함 $y_0y_1=10$ 으로 인한 상황을 고려하면, 이때 입력값에 따른 결과는 $f=g=1$ 을 갖게 되며, 무코드 출력값(non-code output)은 예러를 나타내게 된다. 사실, 회로는 단일 또는 단항성 다중 고장에 대해서 전적으로 자가 검진기의 역할을 한다.[2]



<그림 6> Two-rail 검사기

3. 결론

본 논문은 실시간 기반 결함 허용 시스템을 구성하기 위해 자체-검사 회로를 이용한 고장-안전 시스템의 신뢰성 향상 문제를 다루었다. 이 기술에 의한 안전성은 전체 시스템에서 각 각의 자체-검사 모듈에 결함 허용 기술을 적용하여 안전성과 신뢰성 향상을 TMR과 비교하여 구체화시키는데 큰 의미가 있다. 물론, 동적 여분기법과 정적인 여분기법의 비교 우위에 대한 고찰이라는 면에서는 큰 의미 부여를 할 수는 없지만 기능 신뢰성 면에서 각 모듈에 대한 신뢰성 비교 우위에 대한 고찰에 연구 목적이 있다고 할 수 있다. 고장-안전 제어 인터페이스는 결함 허용 기술을 이용한 자체-검사 모듈들의 신뢰성 향상에 가장 큰 원인이 된다는 것을 알 수 있다. 이런 자체-검사 모듈은 모듈 당 73%의 오버헤드를 나타내며 TMR보다 높은 기능 신뢰성을 갖고 동작하고 고장-안전 시스템은 하드웨어적 비용이나 정확도면에서 앞선다는 것을 알 수 있다. 이러한 기능 논리의 신뢰성 향상을 형식 검증 기법을 이용하여 고장 안전 제어 논리를 검증하는 것은 고장 안전 제어 논리의 핵심 기술인 주기적인 코드 영역의 주입을 통하여 다중 결함 검출을 할 수 있고 회로의 집적화를 이룰 수 있다는 결론을 얻었으며 기존의 BIST 논리 보다 논리 제어 자체가 정보 여분 기법을 이용하기 때문 하드웨어적 부하가 적을 것으로 예상된다. 위와 같은 고장 안전 제어 논리의 개발은 철도, 항공, 선박 등 임계-안전 시스템 구성이 필수적인 분야에서 꼭 필요하며 높은 부가가치와 파급효과가 있다. 보다 나은 함수적 고찰과 검증 기법의 다양화를 통해 보다 많은 산업 분야에서의 임계-안전 시스템 구성이 절실하며 나아가서는 이상적인 신뢰성 구현을 위해 최적화 고장-안전 시스템 구성이 필요하다.

참고문헌

- [1] Barry W. Johnson(1989), *Design and Analysis of Fault-Tolerant Digital Systems*, Addison-Wesley.
- [2] Parag K. Lala(1985), *Fault-Tolerant and Fault Testable Hardware Design*, Prentice -Hall
- [3] Michael Nicolaidis(1998), Fail-Safe Interface for VLSI :Theoretical Foundations and Implementation, *IEEE Computer Society*. Volume 47, No. 1.
- [4] Dmitris Nikolos(1998), Optimal Self-Testing Embedded Parity Checkers, *IEEE* Volume 47, No. 3. .
- [5] TAKASHI NANYA(1988), Error Secure/Propagating Concept and its Application to the Design of Strongly Fault-Secure Processors, *IEEE* and TOSHIAKI KAWAMURA. Volume 37, No. 1.
- [6] Cecilia Metra, Michele Favalli, Piero Olivo, and Bruno(1997), On-Line Detection of Bridging and Delay Faults in Functional Blocks of CMOS Self-Checking Circuits, *IEEE Computer-Aided Design* Volume 16, No. 7.
- [7] Douglas A.Anderson and Gernot Metze(1973), Design of Totally Self-Checking Circuits for m-out-of-n Codes, *IEEE Trans.. Computers*, Volume 22, No.3. 263-269.
- [8] Neil Storey(1996), *Safety-Critical Computer Systems*, Addison-Wesley.
- [9] T.R.N. Rao and E. Fujiwara(1989), *Error-Control Coding For Computer Systems*, Prentice Hall
- [10] W. Carter and P. Schneider, Design of Dynamically Checked Computer, *Proc IFIP Congress*, 878-883 Amsterdam.
- [11] J. Smith and G. Metze(1978), Strongly Fault Secure Logic Networks, *IEEE Trans, Computer*, Volume 27, No. 6, 491-499.
- [12] M. Nicolaidis(1988), A Unified Built-In Self-Test Scheme : UBIST, *Proc. FTCS*, Volume 18, 157-163, Tokyo.
- [13] M. Nicolaidis(1990), Efficient UBIST Implementation for Microprocessor Sequencing Parts, *Proc. Int'l Test Conf*, Volume 21, 316-326, Washington D.C.