

자체시험(Self-Testing) 특성을 갖는 비교기(Comparator) 설계 (The Design of Self Testing Comparator)

양 성 현* 이 상 훈**
(Sung-Hyun Yang) (Sang-Hun Lee)

요 약

본 논문은 NOMS 와 CMOS 집적회로에서 발생가능한 물리적 결점에 의한 결함에 대해서 Fail-safe 시스템에서 사용할 목적이며, 첫 번째 VLSI 회로 상에서 다양한 물리적 결점을 반영할 수 있는 PLA에 대한 결함 모델을 제시한다. PLA에 근거한 설계 이유는 VLSI칩에서의 물리적 결점을 세부적으로 분석하는 것이 너무 복잡하기 때문이다.

두 번째 논문에서는 2단 AND-OR 또는 NOR-NOR 회로로 구현한 설계가 최적의 크기를 갖는다는 것을 보여준다. 또한 NOR-NOR PLA로 구현한 비교기가 제시한 단일 결함 모델에 대해서 자체시험성을 갖는다는 것을 증명한다. 최종적으로 Fail-safe 가산기에 대해 빌딩블럭으로 자체시험 비교기의 적용을 논한다.

ABSTRACT

This paper presents the implementation of comparator which are Fail-Safe with respect to faults caused by any single physical defect likely to occur in NMOS and CMOS integrated circuit. The goal is to use it at the Fail-Safe system. First, a new fault model for PLA(Programmable Logic Array) is presented. This model reflects several physical defects in VLSI circuits. It focuses on designs based on PLA because VLSI chips are far too complex to allow detailed analysis of all the possible physical defects that can occur and of the effects on the operation of the circuit.

Second, this paper show that these design, which was implemented with 2 level AND_ORor NOR-NOR circuit, are optimal in term of size. And it also present a formal proof that a comparator implemented as NOR-NOR PLA, based on these design, is self-testing with respect to most single faults in the presented fault model. Finally, it discuss the application of the self-testing comparator as a building block for implementing Fail-Safe Adder.

1. 서론

이상적인 컴퓨터 시스템은 언제나 정확한 결과만을 출력하는 것이 바람직 하지만 설계 과정에서의 결함, 제작 과정에서의 결점, 기타 하드웨어상의 결함들로 인해 이상적인 시스템 설계는 불가능한 것이 현실이다.

이를 위하여 결함 원인이 될 수 있는 시스템 변화에 적용하여 정확한 출력을 발생하는 확률을 높이고 지속적인 시스템 정상 동작을 성취하게 시도하는 것이 결함허용 시스템이다.[1][2]

* 정회원 : 광운대학교 전자공학부 교수

논문접수 : 2001. 2. 14.

* 종신회원 : 광운대학교 전자공학부 교수

심사완료 : 2001. 2. 23.

※ 본 연구는 '99년도 광운대학교 99년도 교내 연구비에 의해 수행되었음.

이때 컴퓨터 시스템에 대한 최소 요구 조건은 시스템이 부정확한 출력을 정확한 출력으로 판단하지 않게 해야한다는 것이다. 이는 기존 산업분야의 시스템이 컴퓨터화 되어 가고 사무용 시스템의 광역화에 따라 시스템 오류의 결과는 기존의 단순한 불편함을 넘어 상상을 초월하는 경제적 손실과 산업현장에서의 인명 피해까지를 유발하게 되므로써 그 중요성이 강조되고 있다. 특히 비행시스템, 항공관제, 공정관리, 무기 등의 군사장비 같은 기계를 포함하는 복합 시스템의 일부로 컴퓨터가 내장 될 때(Microprocessor Embedded System) 시스템의 오동작으로 인한 결과는 더욱 분명하다[3]. 이러한 시스템을 Safety-Critical 시스템 또는 Safety-related 시스템으로 표현하고 있으며, 이들 중 일부 시스템은 시스템 고장 출력으로 인한 피해가 발생하지 않게 하기 위해서 안전한 상태(Safe State)를 보장 할 수 있는 시스템 출력 상태를 정의하고 있다. 이와 같이 안전측(Safe Side)이 확보된 시스템을 Fail-Safe 시스템으로 표현하며, 기존의 Fail-Safe 시스템의 1차적 목적은 시스템의 RAMS(Reliability, Availability, Maintainability, Safety) 향상을 위한 결함허용(Fault-Tolerant) 시스템으로 설계하는 것이었다. 따라서 결함 검출은 부정확한 출력 수용을 방지 할 뿐만 아니라 결함을 복구하는데 첫 번째 단계가 된다.

이때 결함허용 시스템의 기본이 되는 오류 검출 방법으로 자체검사(Self-checking) 기능을 사용하기도 한다. 그러나 결함허용 시스템은 정해진 시스템 고장 모드에 대해 일정한 수준까지 고장을 검출하고 마스크(Mask) 할 수는 있어도 시스템의 Fail-Safe 동작을 확신 할 수 없으며, 여분(Redundancy)을 이용하기 때문에 경제적 부담이 따른다. 특히 시스템의 성능상 또는 고장 특성상 복구 할 수 없는 고장이 발생했을 때 안전상태로 정의된 출력을 발생하게 하는 고장안전 논리는 안전측이 확보된 Safety-Critical 시스템 영역에서는 필수적이다. 이를 만족하기 위해서 채택하고 있는 방식은 결함허용 시스템으로부터의 출력은 신뢰성이 있는 것으로 가정하고, 시스템 고장이 발생할 때 동작기를 여기 시키지 않게 하기 위해 개방 상태(Open State)로만 고장 발생 특성이 있는 릴레이(Relay) 소자를 활용하고 있다. 이는 시스템에 오류가 존재할 때 그것을 교정하거나 시스템이 안전

측(일반적으로 시스템 동작중지를 의미)으로 동작할 수 있는 Fail-Safe 신호에 의해서 동작기가 제어되어야 함을 의미하며, 고장안전 시스템은 결함 검출 및 결함 발생 시 안전측(Safe Side)으로 동작이라는 두 가지 특성을 만족해야 한다.

오류 검출 방식으로는 Off-line 과 On-line 방식을 고려할 수 있지만 시스템의 가용성을 고려할 때 BIST(Built-In Self-Test)와 같은 Off-line 방식은 알맞지 않다. 이에 반해 검사기(Checker) 회로는 On-line 오류 검출방법으로써 결함 허용 시스템에서 중요한 역할을 한다. 검사기는 시스템내의 정보의 집합이 일정한 코드에 순응하는가를 감시한다. 이때 사용되는 대표적인 코드는 패리티 코드, m-out-of-n 코드, 산술 코드와 2-레일 코드이다.[4]

이것은 결함이 발생하면 감시되는 정보가 더 이상 정해진 코드에 순응하지 않게 변화될 것이라는 가정을 근거로 한다. 따라서 만약 결함이 코드에 순응하는 방식으로 정보의 변화를 가져온다면 이러한 오류는 검사기에 의해 검출할 수 없게 된다.

현대 VLSI 칩에서 단일 고착 결함만을 고려하는 기존의 결함 모델은 더 이상 유효하지 않다. 예를 들면, 하나의 물리적 결점은 여러 선(line)상에서 오류로 귀착되거나 조합회로를 유한 상태 기계(Finite state machine)로 변환할 것이다. 따라서 단일 패리티 비트와 같은 단일 오류 검출 코드로는 VLSI 회로에서 발생가능한 오류를 검출할 수 없으며, 더욱 그와 같은 결함들의 영향을 기존의 시뮬레이팅 방법을 이용하여 예측 할 수 없기 때문에 결함 커버리지 평가가 어렵다. 또한 한 가지 형태의 오류검출 코드로는 어느 것도 마이크로 프로세서와 같은 복잡한 칩에 대해 철저히 오류를 검출하기 위해서 적당하지 않다. 해밍 코드가 레지스터나 버스 전송에서 오류를 검출하는데 이용되는 반면 산술코드는 ALU의 연산을 검사하는데 필요하다. 따라서 칩의 각 부분에 각기 다른 코드를 사용하여 다른 검사기를 구현하는 것은 VLSI 칩 설계를 더 복잡하게 만들며 설계와 설계 검증 과정의 복잡도 증가는 설계가 정확하다는 확신을 감소시켜서 시스템의 전체 신뢰성을 감소시키는 결과가 된다.[5]

이에 반해 중복과 부합(Duplication and Matching) 방법은 논리 연산과 제어선 검사에 알맞으며 현대 VLSI 기술은 중복과 부합 방법이 오류 검출 구현을

위해 유용한 방법이 되게 한다. 중복된 기능 모듈은 병렬로 그 기능을 수행하고 각 모듈 출력을 비교하므로써 오류를 검출한다. 두 개의 모듈중 하나 또는 둘 다 고장이 발생하는 한, 오류는 검출되어 서로 다른 출력을 발생한다.

이 방법에서는 결점이 어떻게 모듈에 영향을 미치는가를 정확하게 예측할 필요가 없고, 설계, 설계 검증, 칩을 시험하는데 과정의 복잡도가 증가하지 않는다. 중복과 부합 기법은 각기 독립된 기능 모듈을 사용하여 공정 과정에서의 결점 또는 검출되지 않는 설계 결함에 기인하는 시스템 고장 문제를 완화하는데 이용될 수 있다.

중복과 부합 체계에서 임계적인 요소는 두 개의 기능 모듈로부터 출력을 비교하는 회로이다. 이러한 비교기에서 검출되지 않는 결함은 기능 모듈들의 출력 불일치를 마스크 한다. 따라서 비교기는 정상동작 동안에 비교기 안에 물리적 결점이 오류 발생 지시로 귀착되어야 하는 자체 시험(Self-Testing) 특성을 가져야 한다.[6]

본 논문은 VLSI로 고장 안전(Fail-Safe) 특성 비교기의 설계와 구현을 제시하였다. 대규모 VLSI 칩은 발생할 수 있는 모든 가능한 물리적 결점들의 자세한 해석과 이러한 결함이 회로의 동작에 어떠한 영향을 미치는가를 자세히 해석할 수 없기 때문에 본 논문에서는 PLA(Programmable Logic Array)에 근거한 설계에 초점을 둔다. 다시 말하면 PLA는 VLSI 회로의 간단한 표준 구조이고 보다 철저한 해석에 더 적합하다. 그러한 해석에 근거해서 본 논문에서는 PLA에 대한 새로운 결함 모델을 제시한다. 이 모델은 집적회로에서 발생 가능성이 있음직한 물리적 결함을 반영하였다.

2단 AND-OR 또는 NOR-NOR 회로로 구현한 비교기는 W.C. Carter와 S.L. Wang, A. Avizienis에 의해서 제시되었다. 이를 근거로 NOR-NOR PLA로 구현한 비교기가 새로운 결함 모델에 속하는 단일 결함에 대해서 자체 시험 특성이 있다는 것을 제시하였다.

최종적으로 결함 허용 시스템에 대한 빌딩 블록으로 고장안전 특성 비교기의 응용에 대해서 논한다.

2. 결함(Fault) 모델

자체시험 특성 회로를 설계하고 구현하기 위해서 이용되는 특정 기술에서 발생 가능한 물리적 결점을 고려할 필요가 있다. 일반적으로 디지털 회로 설계는 전압, 전류 레벨에서 보다 부울 논리 레벨에서 수행하기 때문에 물리적 결점이 파악되면 이러한 결점이 논리 레벨에서 회로의 동작에 미치는 영향을 결정하는 것이 바람직하다. 이러한 영향을 수학적으로 표현한 것이 결함 모델이다.

본 장에서는 일반적인 NMOS와 CMOS VLSI회로에 대한 결함 모델을 제시하고 그것들을 이용하여 PLA에 대한 세부적인 결함 모델을 개발한다.

2.1 고착(Stuck-at)결함 모델

가장 간단하고 일반적으로 많이 사용되는 결함 모델은 고착 결함 모델이다. 이 결함은 특정한 신호선, 즉 게이트의 입력이나 출력이 논리적으로 변하지 않고 어떤 논리적인 일정한 값(1 또는 0)을 계속 유지하며, 논리 값이 0이나, 1이냐에 따라서 고착 0 결함과 고착-1 결함으로 나누어진다. 그림의 (a)부분에서 입력 a는 고착-X 결함이 발생한 상태이다. 따라서 입력 a는 X에 따라서 0이나 1의 값을 유지한다. 입력 a에 대해서 고착 0 결함일 경우에는 ϕ_1 으로, 고착 1 결함일 경우에는 ϕ_2 로 두면, NAND게이트의 논리 함수 $F_{\phi_1}(a,b)=1$ 과 $F_{\phi_2}(a,b)=\bar{b}$ 로 표현할 수 있다. 대부분의 물리적인 결함들은 고착 결함 모델로 표현할 수 있지만, 이 모델로 표현치 못하는 결함들도 있다.

2.2 Stuck-open과 Stuck-on 결함 모델

MOS 트랜지스터에서의 일반적인 결함 모델은 Stuck-Open과 Stuck-On 결함 모델이다. 보통의 Stuck-Open결함과 Stuck-On 결함은 동시에 발생하지만, 이 두 가지의 결함을 검출하는 것과 모델링하는 방법은 서로 다르다. 그림1의 NAND 게이트는 각각 두개의 n, p트랜지스터로 이루어진 구조를 가진다. 만약 임의의 트랜지스터에 Stuck-Open 결함이 발생하면 이 트랜지스터는 게이트의 입력 논리값에 상관없이 현

재의 값을 유지하게 된다. 따라서, 전원이나 그라운 드방향으로 전류의 흐름이 발생하지 않게 되는데, 이러한 경우를 고 임피던스(High Impedence)상태라고 한다. 이 경우에 출력 y는 이전의 논리 값을 유지하게 된다. 아래에 <표 1>에 Stuck-Open 결함이 발생하였을 때의 NAND 게이트의 진리표를 보였다. 표에서 $F_{\phi_3}(a,b)$ 는 결함이 발생하였을 때의 논리값이고, M은 이전 상태의 논리 값을 가지고 있다는 뜻이다. Stuck-Open 결함은 이전 상태의 값을 가지게 때문에 모델링시 시간 t를 고려할 필요가 있다.

<표 1> 결함이 발생하였을 때의 NAND gate의 출력 논리표

<Table 1> Faulty Behavior of the NAND Gate

a	b	F(a,b)	$F_{\phi_3}(a,b)$
0	0	0	1
0	1	1	1
1	0	1	M
1	1	0	0

따라서, 결함이 발생하지 않았을 때의 논리 함수는 식(2-1), 결함이 발생하였을 때의 논리 함수는 식(2-2)로 표현된다.

$$F(a,b;t) = \overline{a(t) \cdot b(t)} \quad (2-1)$$

$$F_{\phi_3}(a,b;t) = \overline{a(t) \cdot b(t)} \cdot \overline{(a(t)+b(t))} + F_{\phi_3}(a,b;t-1) \cdot a(t) \cdot \overline{b(t)} \quad (2-2)$$

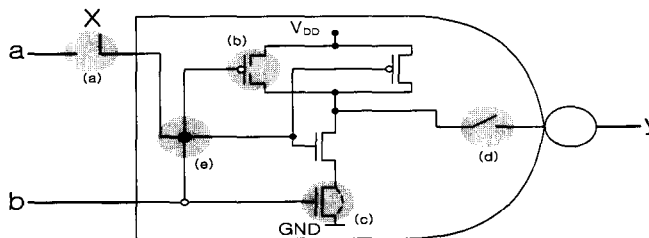
식(2-2)에서 이전 상태를 고려치 않으면, 다음과 같이 표현할 수 있다.

$$F_{\phi_3}(a,b) = \bar{a} \quad (2-3)$$

만약 NAND 게이트의 n-트랜지스터에 Stuck-Open 결함이 발생하였다고 하면, 출력 y는 0값을 가지지 못하기 때문에 고착 1 결함이 발생한 경우와 같다고 볼 수 있다. Stuck-On 결함은 그림1의 (c)부분과 같이 트랜지스터에 단락이 발생한 경우이다. 따라서, 트랜지스터를 영구적으로 동작하는 것처럼 만든다. Stuck-On 결함이 발생한 경우를 ϕ_4 라고 하면, 논리 함수는 $F_{\phi_4}(a,b) = F(a,b)$ 이다. Stuck-On 결함은 신호에서의 지연을 가져오는데 이것을 그림2에 나타 내었다. 그림에서 보듯이 NOT게이트가 5개가 연결 되어 있는데, 이중 첫번째 게이트의 n 트랜지스터에서 Stuck-On결함이 발생하였다고 하자. 결함이 발생 하지 않았다고 하면, 입력 x가 0이면 최종 단의 출력 y5는 1을 출력할 것이다. 하지만, Stuck-On 결함 으로 인해 출력이 왼쪽의 그래프에서와 같이 정확 히 입, 출력되는 것이 아니고, 오른쪽의 그래프와 같 이 지연을 가지고 입, 출력된다. 지연된 신호가 시간 t2에서는 정확한 값으로 인식되지만, 시간 t1에서는 정확한 값을 가지지 못한다.

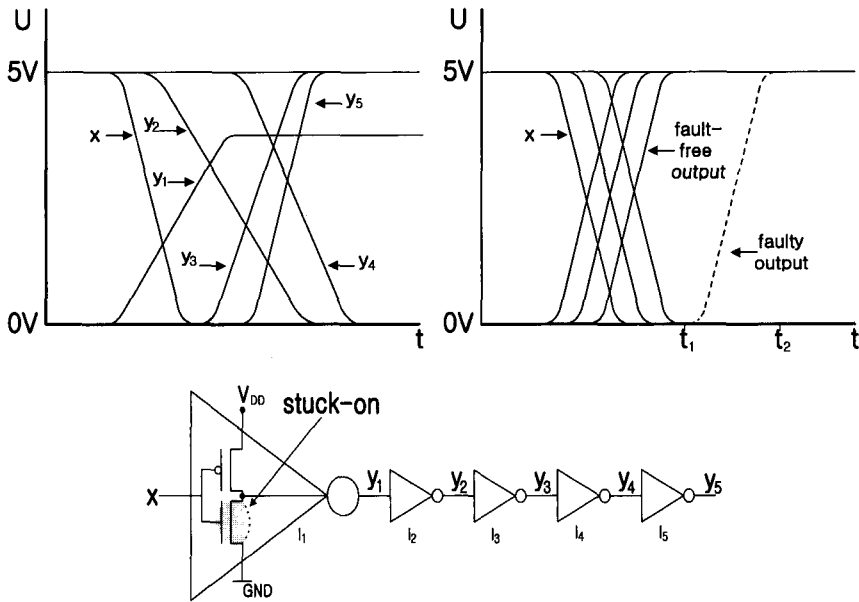
2.3 Open line과 Bridging 결함 모델

Open line 또는 Open 결함은 회로내부의 도선이 Open line 결함은 트랜지스터에서의 Stuck-Open 결함과 같이 고착 X 결함으로 모델링할 수 있다.



[그림 1] CMOS NAND 게이트에서의 여러 가지 결함

[Fig.1] Different Faults in a CMOS NAND Gate



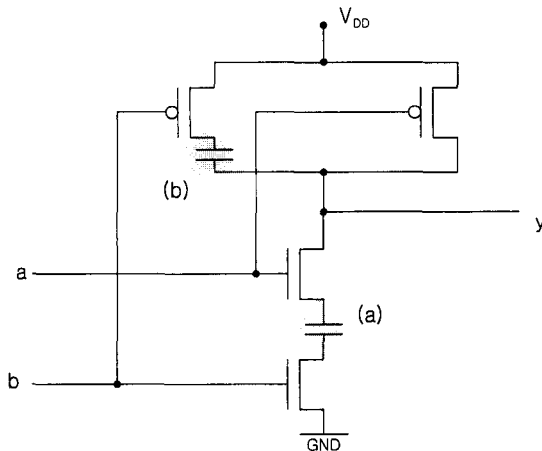
[그림 2] Stuck-On 결함에 의한 신호의 지연

[Fig.2] Influence of a Stuck-on Fault on the Signal Propagation

트랜지스터에서의 Stuck-Open 결함과 마찬가지로 CMOS 게이트에서의 Open Line 결함은 그림 3과 같이 표현할 수 있다. a 부분에서 도선이 단선된다면, 출력 y는 논리 값 0을 가지지 못한다. 따라서 출력 y는 고착 1 결함을 가진다고 볼 수 있다. 그러므로, Open Line 결함의 모델링 방법은 고착 결함, Stuck open 결

함을 모델링하는 방법과 비슷하다고 할 수 있다.

Bridging 결함은 회로 내에서 임의의 도선이 다른 도선과 단락이 된 경우이다. Bridging 결함은 다음의 3가지 경우로 나눌 수 있다.



[그림 3] NAND게이트에서의 Open line 결함

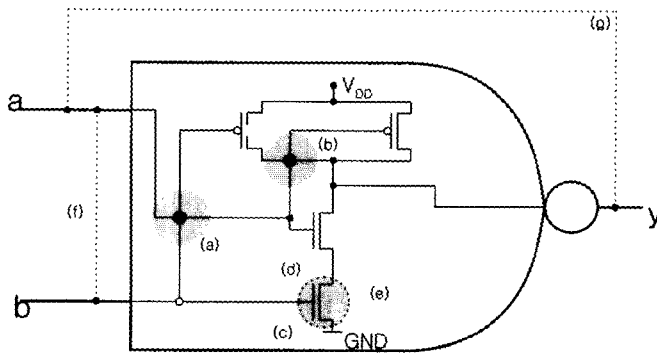
[Fig.3] Open Line Fault in the NAND Gate

- (a) 게이트 내에서의 Bridging 결함
- (b) 페루프(feedback)를 가지지 않는 두 개의 게이트에서의 Bridging 결함
- (c) 페루프(feedback)를 가지는 두 개의 게이트에서의 Bridging 결함

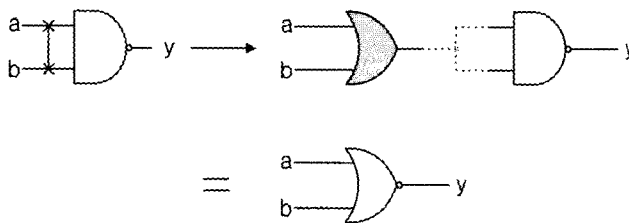
AND)상태로 되어서 NAND 게이트로 입력된다고 볼 수 있다. 이러한 결함을 $\phi 5$ 라 두면, 논리곱의 경우는 $F_{\phi_5}(a,b) = F(a,b)$ 로 논리곱의 경우는, $F_{\phi_5}(a,b) = \overline{a+b}$ 로 표현할 수 있다.

[그림 4]의 (a)부분과 같이 두 입력 사이에 발생한 결함은 게이트 외부에서 입력간에 Bridging 결함이 발생한 (f)부분과 같다고 볼 수 있다. 이 결함은 두 개의 입력이 논리합(Wired-OR)이나 논리곱(Wired-

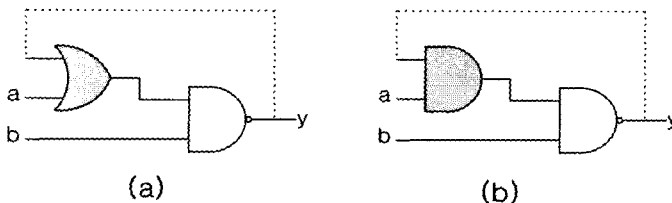
[그림 4]의 (b)경우($\phi 6$)는 (g)경우와 같은 Feedback Bridging 결함이라 할 수 있다. 이 결함은 논리합이나 논리곱으로 표현할 수 있다. 이러한 결함은



[그림 4] NAND게이트에서의 Bridging 결함
[Fig. 4] Bridging Faults in the NAND Gate



[그림 5] Bridging 결함이 발생하였을 때의 동가 회로
[Fig.5] Equivalent Circuit when occurred Bridging Fault



[그림 6] 페루프 Bridging 결함이 발생하였을 때의 동가 회로
[Fig.6] Equivalent Circuit when occurred Feedback Bridging Fault

<표 2> Bridging 결함이 발생하였을 때의 진리표
 <Table 2> Truth Table when occurring Bridging faults

a	b	F	$F_{\varphi_5}^{OR}$	$F_{\varphi_5}^{AND}$	$F_{\varphi_6}^{OR}$	$F_{\varphi_5}^{AND}$	F_{φ_7}	$F_{\varphi_8}^{OR}$	$F_{\varphi_5}^{AND}$	F_{φ_9}
0	0	1	1	1	1	1	1	1	1	1
0	1	1	0	1	u	1	1	1	1	1
1	0	1	0	1	1	1	1	u	1	R
1	1	0	0	0	u	u	1	0	0	0

어떤 입력 조합에 대해서도 출력 y가 발진이 일어나도록 만든다. 나머지 (c), (d), (e) 경우를 $\varphi_7, \varphi_8, \varphi_9$ 라고 두고, <표 2>에 진리표를 나타내었다.

<표 2>에서의 u의 의미는 발진 상태라는 의미이고, R의 의미는 내부 트랜지스터의 저항에 따라 출력 y가 변한다는 의미이다.

2.4 지연 결함(Delay Fault) 모델

지연 결함(Delay Fault)은 신호가 임의의 회로상의 노드에서 정해진 범위의 시간에 도달하지 않을 경우이다. 이 결함을 모델링하기 위해 가장 많이 쓰이는 모델은 단일 게이트 지연(Single Gate Delay) 결함 모델과 경로 지연(Path Delay) 결함 모델이다. 결함 모델을 모델링하기 위해 임의의 회로내의 노드에 신호가 지연을 가지고 입력된다고 하면, 이 노드에서는 현재 이전의 논리값이 유지된다고 볼 수 있다. 따라서, 이러한 지연 결함은 게이트내의 트랜지스터의 Stuck-Open결함과 유사하다고 볼 수 있고, Stuck-Open결함을 지연 결함에서 지연시간이 무한대인 경우라고 볼 수 있다.

3. 비교기 회로 설계

3.1 2 레일(Two-rail) 코드 검사기(Checker)

코드 검사기는 on-line 오류 검출기능을 갖는 컴퓨터 시스템에서 가장 중요한 부분이다. 본 장에서는 VLSI 회로로 자체 시험 특성 비교기를 구현하기에 앞서 2 레일 코드 검사기와 비교기 설계에 대해서

논한다. 두 개의 n 비트 벡터 $A=(a_{n-1}, a_{n-2}, \dots, a_0)$ 와 $B=(b_{n-1}, b_{n-2}, \dots, b_0)$ 가 비교된다고 할 때 $0 \leq i \leq n-1$ 인 모든 i에 대해서 $a_i = b_i'$ 이라면 A, B벡터를 조합한 2n 비트 벡터 $AB = (a_{n-1}, a_{n-2}, \dots, a_0, b_{n-1}, b_{n-2}, \dots, b_0)$ 는 2 레일 코드워드이다. $B' = (b_{n-1}', b_{n-2}', \dots, b_0')$ 일 때 비트 벡터 AB'를 입력으로 갖는 2-레일 코드 검사기는 벡터 A,B의 비교기와 같다. 따라서 모든 입력 비트가 보수(complement)형과 정상(uncomplement)형으로 이용 가능하다는 가정 하에서 비교기 설계와 2-레일 검사기 설계의 차이점은 없다. 오류검출을 위해 중복과 부합(Duplicate and Matching) 방법을 이용할 때 다음과 같은 가정에서 출발한다.

1. 영구 결함과 간헐 결함은 중복된 기능 모듈과 비교기에서 동시에 발생하지 않는다.
2. 기능 모듈 중 하나에서 첫 번째 결함이 발생하호, 그것이 회로 내에서 또는 회로의 상태를 영구적으로 변화시키는 원인이라면, 두 모듈로부터 출력은 불일치하고, 비교기에서 결함이 발생하기 전에 비교기로부터 비코드 출력으로 귀착된다.
3. 비교기나 기능 모듈에서 부가적인 결함이 발생 전에 비교기의 입력이 되는 코드워드의 집합은 비교기의 완전한 자체 시험 특성을 만족한다. 만약 이기간 동안에 결함이 지속되고 비교기가 자체시험 특성을 만족한다면 비교기의 출력은 기능 모듈중 하나의 모듈에서의 결함으로 인한 오류성 기능 출력을 유도하기 전에 오류 발생을 표시할 것이다.

이러한 가정을 근거로 2장에서 정의한 모든 단일 결함을 포함한 결함 모델에 대해서 비교기는 자체 시험 특성이 될 것을 필요로 한다.

본 논문에서는 Carter와 Schneider가 제시한 자체 시험 특성 2-레일 검사기를 근거로 비교기를 논한다. [그림 7]에서 회로는 코드 입력에 대해서 (C1, C0)=(0,1)또는 (1,0) 비코드 입력에 대해서 (C1, C0)=(0, 0)또는 (1, 1)을 갖는 2개의 출력선을 갖는다.

Carter와 Schneider의 검사기는 결함이 없을 때 최소 하나의 코드 입력에 대해서 회로의 모든선은 0 이고, 또 하나는 1이다. 만약 하나의 선이 고착 0 이거나 고착 1 이라면 1이나 0로 생각되는 선에 대한 코드 입력은 출력 (0, 0)또는 (1, 1)이 된다.

Wang과 Avizienis에 의해 제안된 검사기는 식 (3-1)의 합의 곱(sum-of-products)으로 나타낼 수 있다.

입력의 정수 k 개의 대해서 I_k 는 0과 k-1사이의 k 개의 정수 집합이라 놓고, ($I_k = \{0, 1, \dots, k-2, k-1\}$) Q가 집합일 때 |Q|는 Q에서의 집합 원소의 수를 나타낸다.

$$C_0 = \sum_{\{Q|Q \subset I_n, |Q| \text{ even}\}} \{(\prod_{\{i|i \in Q\}} a_i) (\prod_{\{i|i \in \bar{Q}\}} b_i')\} \quad (3-1)$$

$$C_1 = \sum_{\{Q|Q \subset I_n, |Q| \text{ odd}\}} \{(\prod_{\{i|i \in Q\}} a_i) (\prod_{\{i|i \in \bar{Q}\}} b_i')\}$$

3.2 자체 시험 특성 검사기의 최적 설계

Wang과 Avizienis가 설계한 자체 시험 특성 검사기는 n 비트 벡터를 비교하는데 있어서 2n개의 곱의 합을 필요로 했다. 그러나 코드 입력대 대해서 (0, 1)

또는 (1, 0)인 출력하고 비코드 입력에 대해서는 (1, 1)을 출력하는 비교기를 구현하는 것은 다음 방정식으로 가능하다.

$$C_0 = a_0' + b_0' + \sum_{i=1}^{n-1} (a_i b_i' + a_i' b_i) \quad (3-2)$$

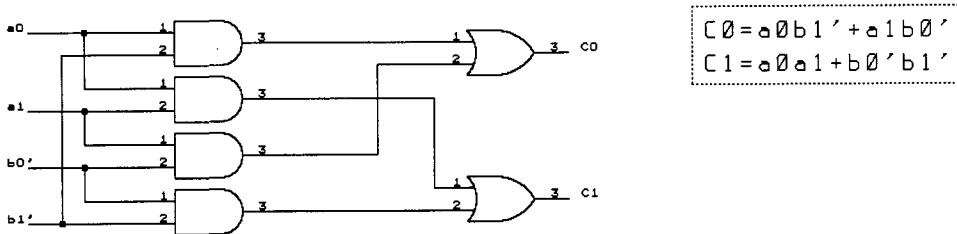
$$C_1 = a_0 + b_0 + \sum_{i=1}^{n-1} (a_i b_i' + a_i' b_i)$$

유사하게 NOR-NOR 형태에서는 식 (3-2)를 근거로 그 함수성을 성취할 수 있다.

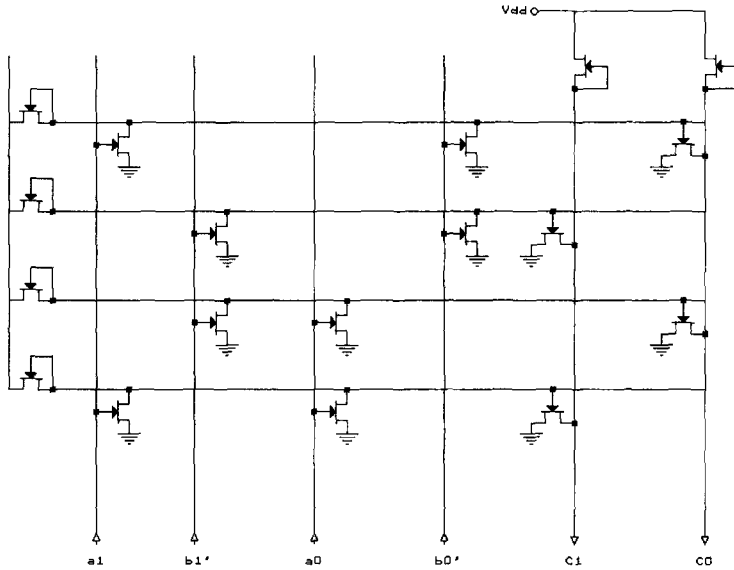
$$C_0 = \text{NOR}_{\{Q|Q \subset I_n, |Q| \text{ odd}\}} \{ \text{NOR}(\{a_i | i \in Q\} \cup \{b_i' | i \in (i_n - Q)\}) \}$$

$$C_1 = \text{NOR}_{\{Q|Q \subset I_n, |Q| \text{ even}\}} \{ \text{NOR}(\{a_i | i \in Q\} \cup \{b_i' | i \in (i_n - Q)\}) \} \quad (3-3)$$

이 비교기는 입력선과 출력선의 결함에 대해서 단지 4n개의 곱의 합으로써 자체 시험 이 있다. 그러나 곱의 합을 만드는 선상에서의 고착 결함에 대해서는 자체시험 특성이 없다. 2ⁿ개의 입력 코드워드 중 각 하나에 대해서 그 코드워드에 의해 1로 선택되는 유일한 곱의 항이 존재한다. 벡터 $A = (a_{n-1}, a_{n-2}, \dots, a_0)$ 의 패리티에 따라서 코드 입력 중 받은 출력 C₀을 선택하고 나머지 받은 출력 C₁을 선택한다. n=2인 경우에 대해서 방정식 (3-3)을 근거로 구현한 NMOS PLA는 그림 8과 같다.



[그림 7] 자체 시험 특성 2-레일 코드 검사기
[Fig.7] A self-testing 2-rail Code Checker



[그림 8] 자체 시험 NMOS 2- 레일 코드 검사기
 [Fig.8] A Self-testing NMOS two-rail Code Checker

4. 결론

본 논문에서는 MOS PLA에 대한 새로운 결함모델을 제시하였으며, 이 모델은 VLSI 회로에서 발생 가능한 여러 결함 형태와 일치한다. 이러한 보다 현실적인 결함모델을 이용하여 NOR-NOR PLA로 구현한 자체시험 특성 비교기는 단일 결함에 대한 고장 안전(Fail-Safe)특성을 갖는다는 것을 알 수 있었다. 또한 기능 모듈 레벨에서 중복과 부합 방법을 이용하여 오류검출을 수행함으로써 비교기의 자체시험 특성을 검증하고 해석하는데서 매우 효과적이었다. 그러나 현재 제한된 시험 입력패턴으로 인해 손실 가능성이 있는 결함 커버리지를 보상할 수 있기 위해서는 설계하고자 하는 회로에 대한 최적의 시험 입력패턴 개발에 대한 연구와 그에 따른 가능성 있는 결함 가이드라인을 설정하는 것에 대한 연구가 수행되어야 할 것이다.

※ 참고문헌

- [1] 양성현, 이기서, "Fault-Tolerance를 위한 시스템 동작 방식에 대한 비교연구," 한국통신학회 논문지, 제17권, 제11호, pp.1297~1289, 1992
- [2] 양성현, 이기서, "TMR 시스템의 설계 및 신뢰도 측정 알고리즘," 대한 전기 학회 논문지 제43권, 제3호, pp.515~527, 1994
- [3] Paray K. Lala, Fault Tolerant and Fault Testable Hardware Design, Prentic Hall International, Inc., London, 1985
- [4] Dhiraj K. Pradhan, Fault-Tolerant Computer System Design, Prentic Hall PTR, 1996
- [5] 양성현, 이기서, "제어 가능한 자체검사 특성 검사기 설계," 한국통신학회 논문지, 제23권, 제5호, pp.1149-1159,1998
- [6] J.P. Khakbaz, "Totally Self-Checking Checker for 1 out of n Code Using Two-Rail Codes," IEEE Tran. on Computer, Vol. C-31, No. 7, pp. 677-681, July, 1982

- [7] S. Kundu, S.M. Reddy, "Embedded Totally Self-Checking Checkers: A Practical Design," *IEEE Design & Test of Computer*, pp.5-12, August, 1990
- [8] Wakerly, *Error Detecting-Codes, Self-Checking Circuit and Applications*, New York, North-Holland, 1978.
- [9] Javad Khakhaz, "Totally Self-Checking Checker for 1-out-of-n Code using Two-Rail codes," *IEEE Tran. on Computers*, Vol.C-31, No.7, pp.677~681, July. 1982.

양 성 현



1958년 2월 1일 생
1983년 2월 광운대학교
전기과 졸업 (공학사)
1987년 8월 광운대학교 대학원
전기과 졸업 (공학석사)
1992년 2월 광운대학교 대학원
전기과 졸업 (공학박사)
1996-1998년: Boston University,
Reserch scientist
1991년 ~ 현재 광운대학교
전자공학부 교수
E-mail : leesh58@mail.gwu.ac.kr

이 상 훈



1958년 8월 2일 생
1983년 2월 광운대학교
응용전자과 졸업 (공학사)
1987년 8월 광운대학교 대학원
전자과 졸업 (공학석사)
1992년 2월 광운대학교 대학원
전자과 졸업 (공학박사)
1991년 ~ 현재 광운대학교
전자공학부 교수
E-mail : leesh58@mail.gwu.ac.kr