

Time-skews 를 고려한 CMOS 회로의 테스트 생성 알고리즘

이철원, 한석봉, 김유홍, 정준모, 선신구, 임인철
한양대학교

Test Generation Algorithm for CMOS Circuits considering Time - skews

C.W. Lee, S.B.Han, Y.H.Kim, J.M. Jung, S.K. Sun, I.C. Lim
Hanyang University

Abstract

This paper proposes a new test generation algorithm to detect stuck-open faults regardless of time-skews in CMOS circuits.

For testing for stuck-open faults regardless of time-skews, in this method, Hamming distance between the initialization pattern and the test pattern is made 1 by considering the responses of the internal gates. Therefore, procedure of the algorithm is simpler than that of the conventional methods because the line justification is unnecessary. Also, this method needs no extra hardware for testability and can be applied to random CMOS circuits in addition to two-level NAND - NAND CMOS circuits.

I. 서론

LSI/VLSI 기술이 발전함에 따라, 비교적 낮은 전력 소모, nMOS 에 비해 빠른 동작 속도, 그리고 고 집적도의 특성으로 인하여 CMOS가 VLSI의 중요한 구성 소자의 하나로 등장하고 있으며, 앞으로 그 사용 범위는 더욱 확대될 전망이다.[1-2]

그러나, CMOS 회로에서는 트랜지스터가 출력 “개방” 상태로 되어 stuck-open (이하 s-op) 고장이 발생할 수 있고, 이러한 고장이 발생할 경우 출력은 CMOS 회로의 전하 저장 기능으로 인하여 전 상태를 그대로 유지하게 되므로 조합 논리 회로가 순서 논리 회로와 같은 동작을 하게 된다. 따라서, 이와 같은 s-op 고장을 검출하기 위해서는 하나의 테스트 패턴으로는 불가능하고, 두 개의 연속적인 테스트 시퀀스가 필요하게 된다.[3-4]

또한, 경우에 따라서 입력 패턴이 다른 패턴으로 천이할 때, 패턴 전체가 동시에 천이하지 않거나, 또는 회로의 경로의 길이가 다르기 때문에 발생하는 unequal delay 등에 의한 Time-skews 가 발생하면, 기존의 패턴으로는 CMOS 회로의 s-op 고장을 검출할 수가 없다.[6-9]

이러한 문제점을 해결하기 위하여, S.M.Reddy 등 [6-7]은 부가 제어 입력과 부가 회로를 이용한 Testable Design 과 고장 회로의 출력에서 static 0-hazards 가 발생하지 않는 테스트 생성 알고리즘을 제안하였다. 그러나, 위의 방법은 테스트를 위한 부가 회로가 필요하고, 2-레벨의 NAND-NAND CMOS 논리 회로에서만 적용이 가능할 뿐만 아니라, line justification 과정으로 인하여 알고리즘 수행 과정이 복잡하다.

본 논문에서는 해밍 거리(Hamming distance)를 사용하여 Time-skews 와 무관하게 CMOS 회로의 s-op 고장을 검출할 수 있는 새로운 결함 시퀀스 생성 알고리즘을 제안한다. 이 알고리즘은 테스트를 위한 부가 회로가 요구되지 않고, 2-레벨의 NAND-NAND CMOS 논리 회로뿐만 아니라 임의의 모든 CMOS 논리 회로에도 적용이 가능하며, line justification 과정을 제거함으로써 기존의 방법보다 알고리즘 수행 과정이 간단하다.

II. CMOS 회로의 고장 모델과 그의 검출

그림 1은 CMOS 회로의 Block Diagram을 나타낸 것이다.

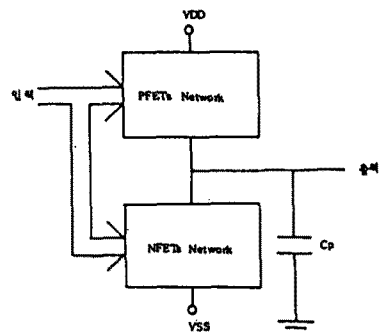


그림 1. CMOS 회로의 Block Diagram

CMOS 회로는 부하 회로(load circuit)인 P-형 FET들 (PEFTs) 과 구동 회로(driver circuit)인 N-형 FET들

(NFETs)로 구성되어 있다. 모든 입력 신호는 트랜지스터의 'gate' 단에 연결되어 있고, 입력신호의 상태에 따라 트랜지스터는 개방 또는 단락된다. 즉, 입력이 "0(1)" 상태이면 PFETs(NFETs)회로의 트랜지스터는 단락되고, NFETs(PFETs)회로의 트랜지스터는 개방된다. 따라서, 출력과 VDD 사이의 경로가 활성화되면 출력과 VSS 사이의 경로가 활성화되지 않으므로 출력은 "1" 상태가 되고, 출력과 VSS 사이의 경로가 활성화되면 출력과 VDD 사이의 경로가 활성화되지 않으므로 출력은 "0" 상태가 된다. 또한, 출력과 양쪽의 source(VDD와 VSS) 간의 경로가 동시에 활성화되지 않으면 출력은 "고 임피던스" 상태가 된다.

위와 같은 경우를 s-op고장이라 하고, 이러한 고장이 발생하면 출력은 CMOS 회로의 전하 저장 기능으로 인하여 전 상태값을 그대로 유지하게 되므로 조합 논리 회로가 순서 논리 회로와 같은 동작을 행하게 된다. 따라서, CMOS 회로의 s-op고장을 검출하기 위해서는 하나의 테스트 패턴으로는 불가능하고, 두 개의 연속적인 테스트 시퀀스가 필요하게 된다.

이러한 고장을 검출하기 위해서, 본 논문에서는 표 1의 규칙에 따라 CMOS 회로를 게이트 레벨로 모델화하고, 표 2와 표 3을 이용하여 다음 테스트 패턴 생성 알고리즘을 수행한다.

<<테스트 패턴(T2)생성 알고리즘>>

단계 1 : 표 5의 규칙에 따라 CMOS 회로를 게이트 레벨로 모델화한다.

단계 2 : 현재 고장 리스트(Curren Fault List: CFL)을 만든다. CFL은 테스트할 트랜지스터의 집합을 말한다.

단계 3 : CFL의 첫번째 트랜지스터를 선택하여 표 6에 따른 논리 회로에서의 등가 고장을 고려하고 D-알고리즘을 이용하여 테스트 패턴을 생성한다. 이 트랜지스터를 CFL에서 제외한다.

단계 4 : D-cubes가 전파되는 경로상에서 단계 3에 의해 생성된 테스트 패턴으로 검출할 수 있는 트랜지스터를 구한다. 이 트랜지스터들을 CFL에서 제외한다. N-part(P-part)에서의 s-op고장을 검출하기 위한 테스트 패턴은 다음 단의 P-part(N-part) 경로를 활성화한다.

단계 5. 다음 과정에 의해서 D-cubes가 전파되는 경로 이외의 경로를 활성화하는 최종적인 테스트 패턴을 구한다.

단계 5.1 : P-part(N-part)에서의 'gate' 입력단이 0(1)로 할당되었으면 단계 5.2로 가고 아니면 단계 6.0으로 간다.

단계 5.2 : 해당 게이트의 또 다른 'gate' 입력단이 don't care로 할당되었으면 단계 5.3으로 가고 아니면 단계 5.4로 간다.

단계 5.3 : OR(AND) 게이트인 경우, don't care를 0(1)로 할당한다. 해당 트랜지스터들을 CFL에서 제외한다. 단계 6.0으로 간다.

단계 5.4 : OR(AND) 게이트인 경우, 해당 게이트의 또 다른 'gate' 입력단이 0(1)로 할당되었으면 단계 3에서 구한 테스트 패턴으로 해당 트랜지스터의 s-op고장을 검출할 수 있다. 이 트랜지스터들을 CFL에서 제외한다.

단계 6 : CFL이 공집합이면 단계 7로 가고 아니면 단계 3으로 간다.

단계 7 : 구한 테스트 패턴을 두 개의 부분 집합 P0, P1으로 나눈다. 여기서, P0, P1은 최종 게이트 응답이 "0" 또는 "1"에 따른다.

단계 8 : 종료.

그림 3은 그림 2의 CMOS 회로를 게이트 레벨로 모델화한 것이고, 표 4는 그림 2의 CMOS 회로에 본 알고리즘을 적용한 결과이다.

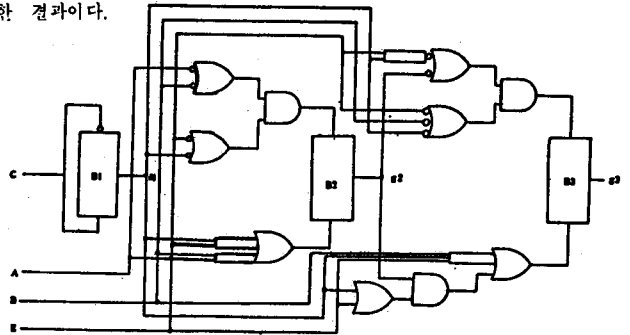


그림 3. 그림 2의 CMOS 회로에 대한 게이트 레벨 모델화

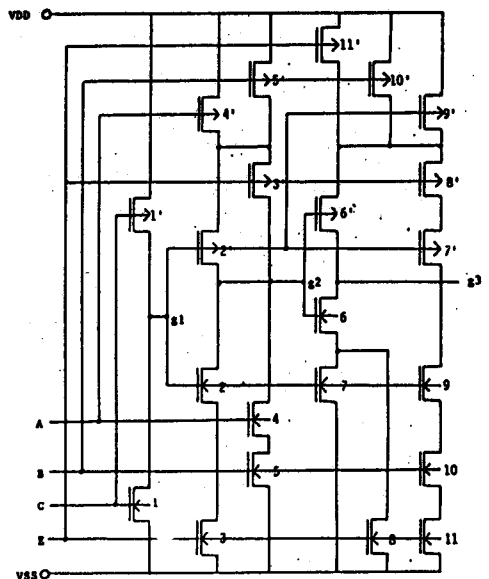


그림 2. CMOS 회로 : $f(A, B, C, E) = \bar{B}\bar{C}E + ABC + AB\bar{E} + \bar{C}\bar{E}$

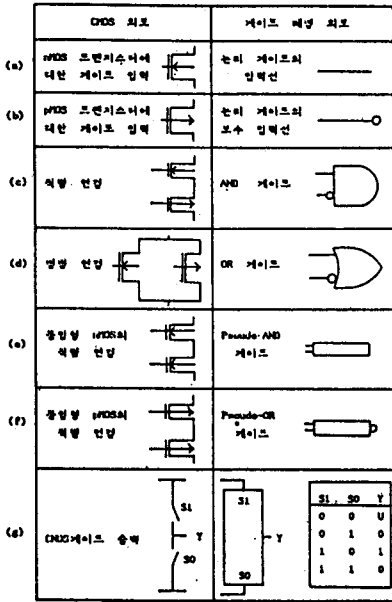


표 1. CMOS 회로를 게이트 레벨로 모델화하기 위한 규칙

CMOS 회로의 고장	논리 게이트의 고장
입력단의 s-a-1 (s-a-0) 고장	입력단의 s-a-1 (s-a-0) 고장
출력단의 s-a-1 (s-a-0) 고장	출력단의 s-a-1 (s-a-0) 고장
pMOS open 고장	입력선의 s-a-1 고장
pMOS on 고장	입력선의 s-a-0 고장
nMOS op 고장	입력선의 s-a-0 고장
nMOS on 고장	입력선의 s-a-1 고장

표 2. 등가 회로

S1	S0	출력
x	1	0
1	0	1
0	0	U

U: Unknown 상태

(a) Singular cover

S1	S0	출력
1	D	\bar{D}
1	\bar{D}	D
D	\bar{D}	D
\bar{D}	D	\bar{D}
0	D	\bar{D}
0	\bar{D}	D
D	0	D
\bar{D}	0	\bar{D}
D	D	\bar{D}
\bar{D}	\bar{D}	D

(b) D-cubes

표 3. 불리의 Singular cover 와 D-cubes

	테스트패턴	내부 게이트 답			완성화된 트랜지스터		
		ABCE	g1	g2	g3	g1	g2
P0	1011	0	1	0	1	2',5'	6,8
	x101	-	-	0	-	-	9,10,11
	0100	-	1	0	-	3',4'	6,7
P1	1001	1	0	1	1'	2,3	6',10'
	1100	-	0	1	-	4,5	6',11'
	1111	-	0	1	-	-	6',9'
	xx10	-	-	1	-	-	7',8'

표 4. 그림 2에 본 알고리즘을 적용한 결과

III. Time-skews 를 고려한 CMOS 회로의 Stuck-open 고장 검출

S-op 고장과 같은 비 고전적인 고장의 검출에 있어서는 테스트 집합뿐만 아니라 테스트 시퀀스에 대한 고려가 필요하다. 즉, 테스트 입력 패턴이 다음 패턴으로 바뀔 때 동시에 바뀌지 않거나, 또는 회로의 경로의 길이가 다를 때 문에 생기는 unequal delay 등에 의하여 Time-skews 가 발생할 때는 s-op 고장을 검출할 수 없다.

예로써, 그림 4 와 같은 회로에서 트랜지스터 5'에 s-op 고장이 발생하였다고 가정하자. 초기화 패턴 T1은 5'이 P-part에 있으므로 출력 f를 "0"으로 만드는 테스트 집합 $\{(1,0,0,1), (0,1,0,1), (0,1,1,0)\}$ 중 어느 것이나 가능하다.

테스트 패턴 T2는 회로가 무고장 상태일 때는 출력 f를 "1"로 만들고, 5'에 s-op 고장이 발생했을 때는 출력 f를 고 임피던스 상태로 만드는 입력은 $(1,0,1,0)$ 뿐이다. 먼저 T1으로 $(1,0,0,1)$ 을 선택하면, $T1 \dots T2 ((1,0,0,1) \dots (1,0,1,0))$ 로 천이했을 때 입력 (A,B,C,E)는 입력 변화의 Time-skews로 인하여 가상적인 중간 입력 $(1,0,0,0)$ 이 존재 할 수 있다. (이 경우는 입력 E가 입력 C보다 먼저 천이했을 경우에 발생한다) 입력 (A,B,C,E) = $(1,0,0,0)$ 일 때 이 입력은 PFET-7과 PFET-8의 트랜지스터를 활성화하여 출력이 "1"이 되므로 5'의 s-op 고장 검출이 불가능하게 된다. 마찬가지로, $T1 = (0,1,0,1)$, $T2 = (1,0,1,0)$ 인 경우에도, 입력 B가 입력 A보다 먼저 천이하여 생긴 가상적인 입력 $T3 = (0,0,1,0)$ 가 존재하여 5'의 s-op 고장 검출이 불가능하게 된다. 또한, $T1 = (0,1,1,0)$, $T2 = (1,0,1,0)$ 인 경우에도, 입력 B가 입력 A보다 먼저 천이하여 생긴 가상 입력 $T3 = (0,0,1,0)$ 으로 인하여 5'의 s-op 고장 검출이 불가능하게 된다.

이상과 같이, Time-skews 를 고려하지 않을 때 5'의 s-op 고장 검출이 가능한 3가지 테스트 시퀀스가 Time-skews 를 고려했을 경우에는 모두 s-op 고장 검출이 불가

능하게 된다.

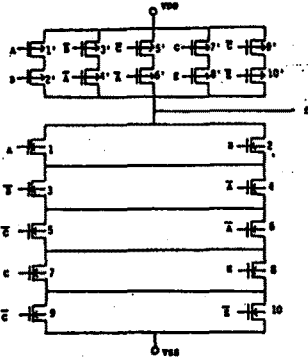


그림 4. CMOS 회로: $f(A, B, C, E) = \overline{A}B + AB + AC + \overline{C}E + CE$

이러한 Time-skews 문제를 해결하기 위하여, 본 논문에서는 다음의 정의 1과 정리 1을 도입한다.

정의 1 : 두 개의 입력 벡터 X와 Y 간의 해밍 거리(Hamming distance) $D(X, Y)$ 를 두 입력 벡터의 대응되는 성분중 서로 다른 성분 쌍의 개수로 정의한다.

정리 1 : CMOS 회로의 s-op고장을 검출하기 위한 결합 시이퀀스에서 내부 응답을 고려한 초기화-테스트 패턴 쌍(T1, T2)에 대한 해밍 거리(Hamming distance)가 1이면, 그 결합 시이퀀스는 Time-skews가 존재하여도 s-op고장 검출이 가능하다.

증명 : 초기화-테스트 패턴 쌍(T1, T2)에 대한 해밍 거리(Hamming distance)가 1이면, 초기화 패턴에서 테스트 패턴으로 천이하는 동안 중간 입력이 존재하지 않는다. 즉, CMOS 회로에서 가상 확장 경로가 발생하지 않는다. 따라서 Time-skews가 존재하여도 초기화-테스트 패턴 쌍에 대한 해밍 거리(Hamming distance)가 1인 결합 시이퀀스는 s-op고장 검출이 가능하게 된다. (증명 끝)

IV. 결합 시이퀀스 생성 알고리즘 및 적용 예

다음 알고리즘은 CMOS 회로에서 Time-skews와 무관하게 s-op고장을 검출하기 위한 결합 시이퀀스 생성 알고리즘을 나타낸 것이다.

《결합 시이퀀스 생성 알고리즘》

- 단계 1. CMOS 회로를 1로부터 시작해서 2-레벨로 분할하여 마지막 단계까지 다음 과정을 수행한다.
- 단계 2. 테스트 패턴 생성 알고리즘에 의해서 생성된 P0(P1)에서 분할한 처음단의 트랜지스터를 활성화하는 테스트 패턴을 선택한다.
- 단계 3. 테스트 패턴에 don't care가 존재하면 단계 5로 가고, 아니면 단계 4로 간다.
- 단계 4 : 다음 과정에 의해 초기화 패턴을 구한다.
 - 단계 4.1 : 테스트 패턴에 대한 내부 게이트 응답이 보수

가 되는 입력 집합(P2)를 구한다.

- 단계 4.2 : 테스트 패턴과 해밍 거리(Hamming distance)가 1이 되는 입력(P3)을 P2에서 구한다.
- 단계 4.3 : 각 내부 게이트에 대한 Boolean 함수를 구한다.
- 단계 4.4 : 분할한 최종 단의 Boolean 함수를 검토하여 입력변화를 결정한다. 즉, 분할한 최종 단의 Boolean 함수에 존재하지 않는 입력 변수가 변한다.
- 단계 4.5 : 단계 4.4의 조건을 만족하는 초기화 패턴을 P3에서 구한다. 단계 6으로 간다.

단계 5 : 다음 과정에 의해 초기화 패턴과 테스트 패턴을 결정한다.

- 단계 5.1 : 내부 게이트 응답을 테스트 패턴에 대한 내부 게이트 응답으로 고정한다.
 - 단계 5.2 : 테스트 패턴에 대한 분할한 최종 단의 응답이 보수가 되는 입력 집합(P2')을 구한다.
 - 단계 5.3 : 각 내부 게이트에 대한 Boolean 함수를 구한다.
 - 단계 5.4 : 분할한 최종 단의 Boolean 함수를 검토하여 입력변화를 결정한다. 즉, 내부 게이트 응답은 변하지 않고 고정되므로 분할한 최종 단의 Boolean에서 don't care가 아닌 입력 변수가 변한다.
 - 단계 5.5 : 단계 5.4를 만족하는 초기화 패턴을 P2'에서 구하고 최종적인 테스트 패턴을 결정한다.
- 단계 6 : 다음 과정에 의해 결합 시이퀀스를 만든다.
- 단계 6.1 : 초기화 패턴에 대한 게이트 응답과 테스트 패턴에 대한 게이트 응답을 서로 비교하여 보수가 되는지 검토한다.
 - 단계 6.2 : 서로 보수가 되는 경우에는 그 테스트 패턴에 의해 활성화되는 트랜지스터의 s-op고장이 검출된다.

단계 7 : P0(P1)이 공집합이면 단계 8로 가고, 아니면 단계 2로 간다.

단계 8 : 최종단계까지 모두 행하였으면 단계 9로 가고, 아니면 단계 1로 간다.

단계 9 : 종료
본 알고리즘을 그림 2의 CMOS 회로를 예로 들어 설명하면 다음과 같다.

먼저, 그림 7의 회로를 내부 게이트 별로 Boolean 함수를 구하면,

$$g1 = \overline{C}, g2 = g1E + AB, g3 = g2(g1 + E) + g1BE$$

$P0 = \{1001, x101, '00\}, P1 = \{1001, 1100, 1111, xx10\}$ 이다. 1단의 트랜지스터를 활성화하는 테스트 패턴(1011)을 선택한다. 테스트 패턴(1011)에 대한 내부 게이트 응답이 서로 보수가 되는 입력 집합 $P2 = \{0001, 0101, 1001, 1100, 1101\}$

이 된다. P2에서 테스트 패턴과 해밍 거리(Hamming distance)가 1이 되는 입력 P3 = {1001} 이 된다. 따라서, 입력 C 만 변화해야 된다는 조건을 만족하는 초기화 패턴은 (A,B,C,E) = (1001)이다. 단계 6에 의해서 s-op고장을 검출하면,

테스트 패턴	내부 게이트 응답		검출된 트랜지스터	
	g1	g2	g1	g2
1001	1	0	-	-
1011	0	1	1	2',5'
1001	1	0	1'	2,3

본 알고리즘을 그림 2의 CMOS 회로에 적용한 결과는 표 5와 같다.

테스트 패턴	내부 게이트 응답			검출된 트랜지스터		
	g1	g2	g3	g1	g2	g3
1001	1	0	1	-	-	-
1011	0	1	0	1	2',5'	6,8
1001	1	0	1	1'	2,3	6',10'
1010	0	1	1	-	-	-
1011	0	1	0	-	-	6,8
1010	0	1	1	-	-	7',8'
1001	1	0	1	-	-	-
1101	1	0	0	-	-	9,10,11
0100	1	1	0	-	-	-
1100	1	0	1	-	4,5	6',11'
0100	1	1	0	-	3',4'	6,7
0111	0	1	0	-	-	-
1111	0	0	1	-	4,5	6',9'

표 5. 그림 2에 본 알고리즘을 적용한 결과

V. 결 론

본 논문에서는 CMOS 회로의 s-op고장 검출을 위한 테스트 생성 알고리즘을 제안하였다.

Time-skews가 발생하면, CMOS 회로에서 s-op고장 검출이 불가능함을 보이고, 해밍거리(Hamming distance)를 도입하여 Time-skews 문제를 해결하였다. 즉, 내부 게이트 응답을 고려하여 초기화 패턴과 테스트 패턴 간의 해밍 거리(Hamming distance)를 1로 하여 결함 시퀀스를 생성하였다.

종래의 방법과는 달리, 2-레벨의 NAND-NAND CMOS 논리 회로 뿐만 아니라 임의의 CMOS 논리 회로에도 적용이 가능하며, 테스트를 위한 부가 회로가 요구되지 않고, line justification 과정을 제거함으로써 알고리즘 수행 과정을 간단화하였다.

참 고 문 헌

1. C.Mead and L.Conway, Introduction to VLSI Systems,

Addison-Wesley, 1980.
 2. A.Mukjerjee, Introduction to nMOS & CMOS VLSI Systems Design, Prentice-Hall, 1986.
 3. R.L.Wadsack, "Fault Modeling and Logic Simulation of CMOS and MOS Integrated Circuits", Bell System Tech. Jour., Vol. 57, No.4, pp.1449-1474, May-June 1978.
 4. S.K.Jain and V.D.Agrawal, "Modeling and Test Generation Algorithms for MOS Circuits", IEEE Trans. on Comp. Vol. C-34, No.5, pp.426-433, May 1985.
 5. M.A.Breuer and R.L.Harrison, "Procedures for Elimination Static and Dynamic Hazards in Test Generation", IEEE Trans. on Comp. Vol.C-23, No.10, pp.1069-1078, Oct. 1974.
 6. S.M.Reddy, M.K.Reddy and J.G.Kuhl, "On Testable Design for CMOS Logic Circuits", International Testing Conf., pp.435-445, Oct. 1983.
 7. S.M.Reddy, M.K.Reddy and V.D.Agrawal, "Robust Tests for Stuck-open Faults in CMOS combinational Logic Circuits", The 14th Inter. Symp. Fault-Tolerant Computer, pp.44-49, 1984.
 8. N.K.Jha and J.A.Abraham, "Testable CMOS Logic Circuit under Dynamic Behavior", IEEE International Conf., on Computer-Aided Design, pp.131-133, Nov. 1984.
 9. S.M.Reddy and M.K.Reddy, "Testable Realization for FET Stuck-open Faults in CMOS Combination Logic Circuits", IEEE Trans. on Comp., Vol. C-35, No.8, Aug. 1986.
 10. J.P.Roth, W.G.Bouricius and P.R.Schneider, "Programed Algorithms to Compute Tests to Detect and Distinguish between Failures in Logic Circuits", IEEE Trans. on Elect. Comp., Vol. EC-16, No.5, pp.547-580, Oct. 1967.
 11. W.W.Peterson and E.J.Weldon Jr., Error-Correcting Codes, 2nd ed., MIT Press, Combridge, Mass., 1972.
 12. 이철원, 이재민, 한석봉, 김윤홍, 임인철, "CMOS 회로의 Stuck-open고장을 위한 테스트 생성", 1986년도 반도체 및 CAD 연구회 합동 학술 발표회 논문집, Vol. 5, No.1, pp.44-47, 1986. 5.