

論文 2003-40SD-12-9

CMOS VLSI에서 트랜지스터 합선 고장을 위한 효율적인 등가 고장 중첩 알고리즘

(Efficient Equivalent Fault Collapsing Algorithm for Transistor Short Fault Testing in CMOS VLSI)

裴 晟 桓 *

(Sung Hwan Bae)

요 약

IDDQ 테스트는 CMOS VLSI 회로의 품질 및 신뢰성 향상에 중요한 테스트 방식이다. 그러나 상대적으로 느린 IDDQ 테스트를 위해서는 고려한 고장 모델에서 발생 가능한 고장의 수를 감소하거나 가능한 적은 수의 테스트 패턴을 유지하는 게 필요하다. 본 논문에서는 IDDQ 테스트에 자주 이용되는 트랜지스터 합선 고장 모델에서 발생 가능한 고장의 수를 효과적으로 감소시킬 수 있는 효율적인 등가 고장 중첩 알고리즘을 제안한다. ISCAS 벤치마크 회로의 모의 실험을 통하여 제안된 방식의 우수한 성능을 확인하였다.

Abstract

IDDQ testing is indispensable in improving quality and reliability of CMOS VLSI circuits. But the major problem of IDDQ testing is slow testing speed due to time-consuming IDDQ current measurement. So one requirement is to reduce the number of target faults or to make the test sets compact in fault model. In this paper, we consider equivalent fault collapsing for transistor short faults, a fault model often used in IDDQ testing and propose an efficient algorithm for reducing the number of faults that need to be considered by equivalent fault collapsing. Experimental results for ISCAS benchmark circuits show the effectiveness of the proposed method.

Keyword : Fault Testing

I. 서 론

CMOS 회로의 크기와 복잡도가 증가하면서 회로가 정상적으로 동작하는지 여부를 검증하는 테스트는 매우 어렵고 많은 시간과 비용을 필요로 하게 되었다^[1]. 따라서 CMOS 회로에서 발생 가능한 다양한 형태의 물리적인 결함을 테스트 대상 회로의 출력 단에서 단

지 논리적인 값을 검사함으로써 결함을 검출하기는 어렵다. IDDQ(quiescent power supply current) 테스트는 CMOS에서 발생 가능한 여러 종류의 물리적 결함을 효율적으로 검출 할 수 있는 테스트 방식이다^[1,2]. 테스트 회로에 존재하는 결함이나 장애의 영향을 기술하기 위해서 사용되는 고장 모델은 실제적인 장애를 정확하게 모델링 해야 되고 고장 집합의 크기가 적당하여 복잡한 시스템도 다룰 수 있어야 한다^[3].

기존의 논리 테스트에 이용되는 고착 고장(stuck-at-fault) 모델은 CMOS 회로에서 발생 가능한 많은 물리적인 결함을 검출하는데 어려움이 있어,

* 正會員, 漢麗大學校 멀티미디어情報通信工學科
(Dept. of Multimedia, Information and Telecommunication Eng., Hanlyo Univ.)
接受日字:2003年5月26日, 수정완료일:2003年11月24日

IDDQ 테스트에서는 게이트 옥사이드 단락 고장 모델 (gate-oxide short fault model)^[1], 의사 고착 고장 모델 (pseudo stuck-at-fault model)^[4], 합선 고장 모델 (bridging fault model)^[5], 트랜지스터 합선 고장 모델 (transistor short fault model)^[6] 등을 이용한다.

IDDQ 테스트 방식은 높은 고장 및 결함 검출률을 갖지만 상대적으로 출력 단에서 논리적인 값을 검사하는 전압 테스트 방식에 비해서 느린 테스트 시간을 가진다. 이러한 문제를 해결하기 위해서 내장된 전류 테스트 (Built-in current testing)을 이용하는 방식이 가능하지만 내장된 감지기(BICS: Built-In Current Sensor)의 오버헤드와 내장된 회로의 테스트에 관련하여 부수적인 문제가 발생한다^{[1], [2]}. 따라서 IDDQ 테스트를 이용하여 테스트 대상 회로의 고장 및 결함을 효과적으로 검출하거나 진단하기 위해서는 고려한 고장 모델에서 발생 가능한 고장의 수를 감소하거나 가능한 적은 수의 테스트 패턴을 유지하는 게 필요하다.

Mao와 Gulati에 의해서 소개된 트랜지스터 합선 고장 모델은 IDDQ 정보를 이용하여 CMOS 회로의 고장 검출(fault detection), 고장 위치(fault location), 고장 진단(fault diagnosis)에 효과적인 모델임이 보여졌다^[6]. 그러나 고려되는 트랜지스터 누설 고장(transistor leakage fault)의 수는 트랜지스터 당 6개의 고장으로 해석되기 때문에 매우 크게 된다. 트랜지스터 합선 고장 모델은 다음과 같은 단점을 가지게 된다. 첫째, 고려되는 고장의 수가 크기 때문에 높은 고장 검출률을 유지하면서 가능한 적은 수의 패턴을 생성하기에 어려움이 발생한다. 둘째, IDDQ 정보를 이용하는 대부분의 고장 위치 검출(또는 진단) 방법은 많은 수의 고장이 존재할 경우에는 실제 사용을 위해서 너무 많은 시간을 소비하는 과정을 필요로 하게 된다^{[1], [4]}. 이러한 단점에도 불구하고 트랜지스터 합선 고장 모델은 IDDQ 테스트를 위한 좋은 고장 모델이므로 고려되는 고장의 수를 효과적으로 감소시키는 방법이 필요하다.

본 논문에서는 CMOS 회로에서 발생 가능한 고장 및 결함을 검출하여 테스트 대상 회로의 품질 및 신뢰성을 향상시킬 수 있는 IDDQ 테스트에 적합한 트랜지스터 합선 고장 모델에서 등가 고장(equivalent fault)을 효과적으로 검출함으로써 고려되는 고장의 수를 감소시킬 수 있는 새로운 등가 고장 중첩 알고리즘을 제안한다. 본 논문은 2장에서 고장 모델과 등가 고장 중첩 기법에 관해서 설명하고, 3장에서는 제안된 등가 고장

중첩 알고리즘에 관한 설명과 ISCAS 벤치마크 회로에 대한 모의실험 결과를 검토한다. 마지막으로 4장에서 결론을 맺는다.

II. 고장 모델과 등가 고장 중첩 기법

1. 고장 모델

CMOS 회로는 구조적 특성상 정적상태(quiescent state)에서는 전류가 수 nA로 거의 흐르지 않는다. IDDQ 테스트는 이러한 CMOS 회로의 특성을 이용하여 고장을 검출하는 방법이다. 정적상태에서 합선 결함, 게이트 옥사이드 단락, 기생 트랜지스터 누설, 누설 PN 결함, 개방 결함 등과 같은 물리적인 결함이나 고장으로 인하여 CMOS 회로의 VDD와 GND 사이에 형성된 전류 경로에 흐르게 되는 수 mA의 큰 정적상태 전류 값을 통해서 테스트 회로의 고장을 용이하게 검출할 수 있다.

본 논문에서 가정된 고장 모델은 IDDQ 테스트를 이용하여 CMOS 회로의 고장 검출, 고장 위치, 고장 진단에 효과적인 트랜지스터 합선 고장이다^{[4], [9]}. <그림 1>에는 인버터 G2의 p-트랜지스터 게이트(g)와 소오스(s) 사이에 합선이 발생한 경우의 예를 보인다. 그림에서 인버터 G2의 입력 단 3에 논리 값 "0"을 갖도록 해 준다면 회로의 정적상태에서도 VDD와 GND 사이에 전류의 경로가 형성되어, 회로의 정적상태에서도 많은 양의 전류가 흘러 p-트랜지스터 게이트(g)와 소오스(s) 사이에 발생한 합선 고장을 검출할 수 있다.

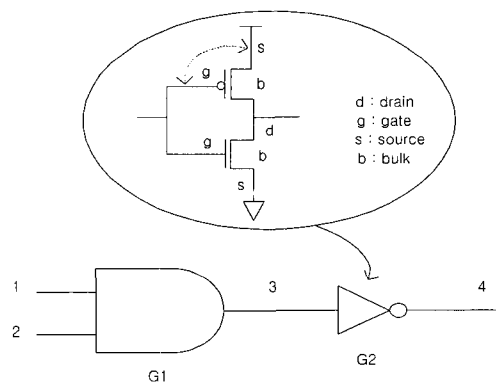


그림 1. IDDQ 테스트의 예
Fig. 1. Example of IDDQ testing.

트랜지스터 합선 고장 모델에서는 테스트 대상 회로

를 게이트 레벨이 아닌 트랜지스터 레벨로 해석해야 한다. 따라서 MOS 트랜지스터의 드레인(drain), 게이트(gate), 소오스(source), 벌크(bulk)의 4개 단자를 고려하기 하기 때문에 <그림 2>에 보인 바와 같이 하나의 MOS 트랜지스터 당 $C_2 = 6$ 개의 가능한 합선 고장이 존재하게 된다.

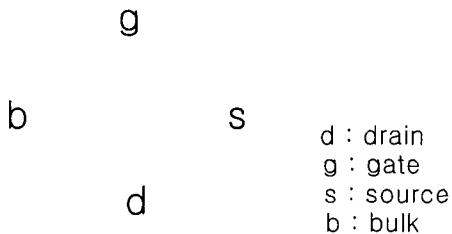


그림 2. 트랜지스터 합선 고장 모델
Fig. 2. Transistor short fault model.

정의 1 : IDDQ 테스트 방식을 이용하여 트랜지스터 합선 고장 f를 검출하기 위해서 테스트 대상 회로에 임의의 테스트 패턴 V를 입력할 경우,

$C(V, f) = 0$, 정상적인 전류 값(normal Iddq), 고장 미 검출

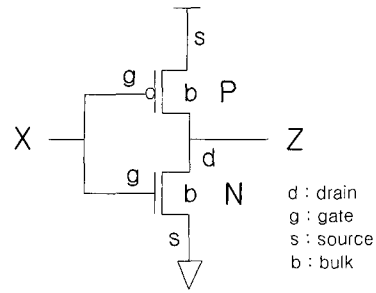
$C(V, f) = 1$, 과도한 전류 값(abnormal Iddq), 고장 검출

정의 2 : IDDQ 테스트 방식을 이용하여 임의의 테스트 패턴 V를 테스트 대상 회로에 적용할 경우에 트랜지스터 합선 고장 f1과 f2가 항상 같은 값의 결과를 나타낼 때, 이를 등가(equivalent)라 한다.

$$C(V, f1) = C(V, f2)$$

트랜지스터 합선 고장 모델에서 등가 고장(equivalent fault)은 IDDQ 테스트 방식을 이용하여 서로 구별할 수 없으므로 등가 고장의 경우에는 단지 하나의 고장만이 고려되고 등가 고장을 검출하는 기술을 등가 고장 중첩이라 부른다^[4,9]. 효율적인 등가 고장 중첩 알고리즘은 가정된 고장 모델에서 IDDQ 테스트로 서로 구별하지 못하는 노드를 삭제함으로써, 고장 시뮬레이션과 테스트 패턴 생성에 요구되는 시간을 효과적으로 줄여주고 가능한 적은 수의 패턴 생성이 가능해진다.

2. 게이트 내부 등가 고장 중첩(intra-gate equivalent fault collapsing)^[6]



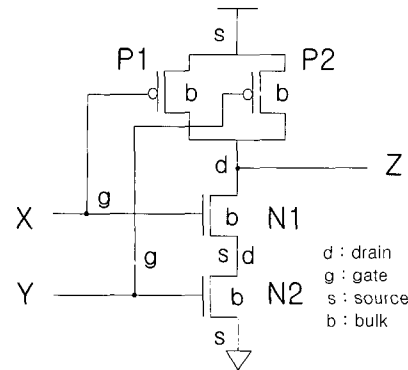
	P						N					
X	bg	bd	bs	sd	gd	gs	bg	bd	bs	sd	gd	gs
0	1	0	0	0	1	1	0	1	0	1	1	0
1	0	1	0	1	1	0	1	0	0	0	1	1

↓

X	f1	f2	f3
0	0	1	1
1	1	0	1

그림 3. 등가 고장 중첩(게이트 내부)을 이용한 인버터의 감소된 고장 표

Fig. 3. Reduced fault table of inverter using intra-gate equivalent fault collapsing.



	P1						P2						N1						N2						
XY	bg	bd	bs	sd	gd	gs	bg	bd	bs	sd	gd	gs	bg	bd	bs	sd	gd	gs	bg	bd	bs	sd	gd	gs	
00	1	0	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0
01	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	1	1
10	0	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	1	0	1	0
11	0	1	0	1	1	0	1	1	0	1	1	0	1	0	0	1	1	1	1	0	0	0	0	1	1

↓

XY	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11
00	0	1	0	0	1	0	0	0	1	1	1
01	0	1	0	1	1	0	1	1	1	0	0
10	1	1	1	0	0	0	0	1	0	1	1
11	1	0	0	0	1	1	1	1	0	0	1

그림 4. 등가 고장 중첩(게이트 내부)을 이용한 2-NAND 게이트의 감소된 고장 표

Fig. 4. Reduced fault table of 2-NAND using intra-gate equivalent fault collapsing.

게이트 내부 등가 고장 중첩 기법은 기본 게이트의 내부를 분석하여 등가 고장을 검출함으로써 트랜지스터 합선 고장 모델에서 초기 기본 게이트가 가지는 고장 표(fault table)의 양을 효과적으로 줄여 테스트 대상 회로에서 고려되는 합선 고장의 수를 줄이는 방법이다. <그림 3>과 <그림 4>에는 인버터와 2-입력 NAND 게이트의 내부 등가 고장 중첩 기법의 예를 보인다. 예를 들어 <그림 3>의 인버터의 경우에 P-트랜지스터 게이트와 벌크간의 합선(bg) 경우에는 입력 단 X에 논리 값 "0"이 적용되면 "1"의 과도한 전류 값을 가지게 되어 고장을 검출할 수 있으며, 입력 단에 논리 값 "1"이 적용되면 "0"의 정상적인 전류 값을 가지게 되어 고장을 검출할 수 없음을 의미한다.

게이트 내부 등가 고장 중첩 기법을 ISCAS 벤치마크 회로에 적용할 경우 트랜지스터 합선 고장 모델에서 고려되는 고장의 수 결과를 <표 1>에 보인다.

표 1. 등가 고장 중첩(게이트 내부)을 이용한 실험 결과

Table 1. Experimental result of intra-gate equivalent fault collapsing.

회로	게이트의 수	고장 수(B)	남은 고장 수(A)	A/B(%)
C880	383	10188	3933	38.6
C1355	546	13080	5700	43.6
C1908	876	16788	6984	41.6
C2670	1193	27488	10258	37.3
C3540	1669	39672	14774	37.2
C5315	2307	60060	22880	38.1
C6288	2416	60672	26576	43.8
C7552	3512	79560	30480	38.3

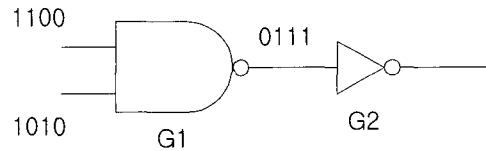
3. 게이트간 등가 고장 중첩(inter-gate equivalent fault collapsing)^[8, 9]

트랜지스터 합선 고장 모델에서 게이트간에 고장을 고려함으로써 더 많은 수의 등가 고장을 검출할 수 있다.

다중 입력 중첩(multiple-input collapsing)

IDDQ 테스트를 적용할 경우에 테스트 대상 회로 안에 존재하는 많은 게이트들은 다중 입력 상황을 고려할 경우에 게이트간에 등가 고장을 검출할 수 있다. <그림 5>에는 2입력 NAND와 인버터로 이루어진 간단한 회로의 예이다. 게이트 G1과 게이트 G2의 감소된

고장 표를 결합할 경우에 추가적인 등가 고장이 검출된다. 여기서 다중 입력의 의미는 하나의 게이트의 모든 입력 단에 논리적인 값을 적용하고 시뮬레이션을 수행한다는 의미이다. 그림에서 게이트 G1의 트랜지스터 고장 f2와 게이트 G2의 고장 f1, 또 게이트 G1의 트랜지스터 고장 f6과 게이트 G2의 고장 f2가 추가적인 등가 고장임을 알 수 있다.



G1	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	G2	f1	f2	f3
00	0	1	0	0	1	0	0	0	1	1	1	1	1	0	1
01	0	1	0	1	1	0	1	1	1	0	0	1	1	0	1
10	1	1	1	0	0	0	0	1	0	1	1	1	1	0	1
11	1	0	0	0	1	1	1	1	0	0	1	0	0	1	1

G1	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	G2	f3
00	0	1	0	0	1	0	0	0	1	1	1	1	1
01	0	1	0	1	1	0	1	1	1	0	0	1	1
10	1	1	1	0	0	0	0	1	0	1	1	1	1
11	1	0	0	0	1	1	1	1	0	0	1	0	1

그림 5. 게이트간 등가 고장 중첩(다중 입력)을 이용한 결합 고장 표의 예

Fig. 5. Example of combined fault table using inter-gate equivalent fault collapsing (multiple input)

단일 입력 중첩(single-input collapsing)

정의 3: 트랜지스터 합선 고장 f를 검출하기 위해서 게이트의 입력 단(a)에 논리 값 "0(1)"을 적용했을 때 IDDQ 테스트 측정값이 다른 입력 단에 값에 관계없이 단일 입력 단(a)에 적용된 값과 항상 같을 경우 합선 고장 f를 양의 값 고장(positive fault)이라 하고, 반대 값의 결과를 나타낼 때, 이를 음의 값 고장(negative fault)이라 한다.

<그림 6>에는 인버터와 2-NAND 게이트의 고장 표에서 양의 값 고장과 음의 값 고장을 보인다. 예를 들어 2-NAND 게이트의 합선 고장 f1과 f9는 입력 단 y에 관계없이 입력 단 x와 관련하여 각각 양과 음의 값 고장이 된다.

x	f1	f2
0	0	1
1	1	0

x y	f1	f9	f7	f10
0 -	0	1	-	-
1 -	1	0	-	-
- 0	-	-	0	1
- 1	-	-	1	0

인버터

2-NAND

그림 6. 양의 값과 음의 값 고장
Fig. 6. Positive and negative faults.

양의 값과 음의 값 고장의 정의를 이용할 경우 게이트간에 단일 입력 상황에서 추가적인 등가 고장이 검출된다. <그림 7>에 보인 간단한 회로의 예에서 게이트 G1의 입력 단 a에 "01"의 값을 적용할 경우, 게이트 G2는 입력 단 c에 관계없이 게이트간 등가 고장 중첩이 가능하다. 즉, 게이트 G1의 고장 f1은 게이트 G2의 고장 f9와 게이트 G1의 고장 f2는 게이트 G2의 고장 f1과 등가 고장이 된다.

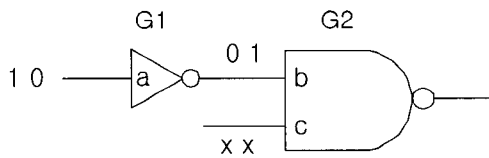


그림 7. 게이트간 등가 고장 중첩(단일 입력)의 예
Fig. 7. Example of inter gate equivalent fault collapsing(single input)

표 2. 등가 고장 중첩을 이용한 실험 결과
Table 2. Experimental result of equivalent fault collapsing.

회로	게이트의 수	고장 수(B)	남은 고장 수(A)	A/B(%)
C880	383	10188	2889	28.4
C1355	546	13080	4006	30.6
C1908	876	16788	4845	28.9
C2670	1193	27488	7314	26.6
C3540	1669	39672	10194	25.7
C5315	2307	60060	16434	27.4
C6288	2416	60672	18523	30.5
C7552	3512	79560	21149	26.6

<표 2>에는 게이트 내부와 게이트간 등가 고장 중첩 기법을 ISCAS 벤치마크 회로에 적용할 경우 트랜지스터 합선 고장 모델에서 고려되는 고장의 수 결과를 보인다.

III. 제안된 등가 고장 중첩 알고리즘과 모의실험 검토

트랜지스터 합선 고장 모델은 CMOS 회로의 고장 검출, 고장 위치, 고장 진단에 효과적인 IDDQ 테스트를 위한 방법으로 테스트 대상 회로를 게이트 레벨이 아닌 트랜지스터 레벨로 해석해야한다. 제안된 등가 고장 중첩 알고리즘은 이러한 트랜지스터 레벨 해석 원리를 이용한 분해와 치환과정을 통해서 더 많은 등가 노드를 효과적으로 검출 할 수 있다.

본 논문에서 제안한 고장 등가 고장 중첩 알고리즘은 <그림 8>에 보인 바와 같이 4 종류의 프로세스로 이루어진다.

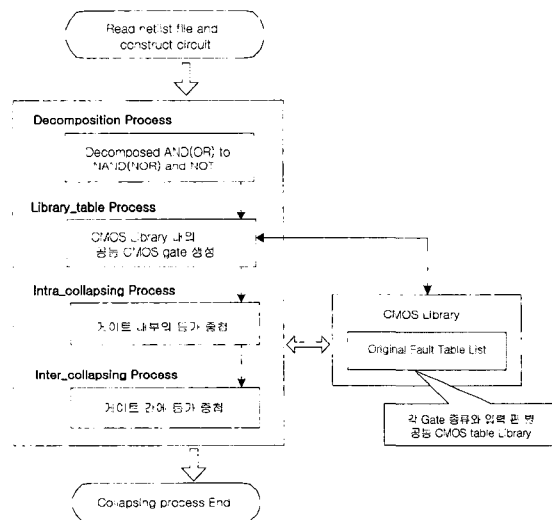


그림 8. 제안된 등가 고장 중첩 알고리즘
Fig. 8. Proposed equivalent fault collapsing algorithm.

1. 재구성 프로세스(Decomposition Process)

재구성 프로세스는 트랜지스터 합선 고장 모델의 처음 과정으로 제안된 알고리즘에서는 AND, OR 게이트는 NAND, OR, 인버터의 구조로 분해하는 방식을 사용한다. 실제로 AND, OR 게이트는 분해가 불가능하지만 트랜지스터 레벨 분석에서는 <그림 9>와 같이 AND는 NAND와 인버터의 구조로 분석 할 수 있다. 이러한 재구성 프로세스를 통해서 다음 단계에서 고려하는 CMOS 기본 게이트들의 라이브러리에 관련한 고장 표의 수가 줄어들며, 또한 Inter_collapsing 프로세스에서 고려하는 게이트간의 등가 고장의 수를 더 검출

할 수 있다.

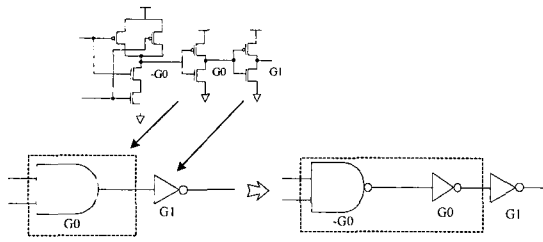


그림 9. 트랜지스터 합선 고장 모델의 재구성 프로세스 예
 Fig. 9. Example of Decomposition Process in transistor short fault model.

2. 라이브러리 표 프로세스(Library_table Process)

라이브러리 표 프로세스는 회로 구성이 끝난 후 각 게이트의 종류와 입력 포트 수에 따라 종류별로 Original Fault Table을 생성하여 CMOS Library의 Original Fault Table List에 저장하여 Intra_collapsing과 Inter_collapsing 프로세스에서 등가 고장 중첩을 수행할 때 각 게이트는 게이트의 형과 입력 포트 수에 맞는 고장 표를 검색하여 복사해서 사용한다.

3. 내부 중첩 프로세스(Intra_collapsing Process)

내부 중첩 프로세스는 트랜지스터 합선 고장 모델에서 재구성 프로세스와 라이브러리 표 프로세스 후에

Gate::Intra_Collapsing_Process

```

...
for(i = 0 ; i < FaultTable.size; i++)
{
    /* 고장 표의 고장을 루프를 돌면서 검사한다 */
    for(j = i+1; j < FaultTable.size; j++)
    {
        /* i번째 fault와 j번째 fault가 같으면 j번째 fault를 제거한다. */
        if(FaultTable[i] == FaultTable[j]) /* i번째 fault와 j번째 fault가 같으면 */
        {
            FaultTable.remove(j); /* j번째 fault를 제거한다. */
        }
    }
}

```

그림 10. 내부 중첩 프로세스의 의사 코드
 Fig. 10. Pseudo-code of Intra_collapsing Process

기본 게이트의 내부를 분석하여 등가 고장 중첩을 검출함으로써, 테스트 대상 회로에서 초기 기본 게이트가 가지는 가능 고장 표의 양을 효과적으로 줄여 고려되는 합선 고장 수를 줄이는 방법이다. 본 논문에서는 게이트 재구성 프로세스 과정 후에 내부 중첩 프로세스를 실행하기 때문에 고려되는 CMOS 기본 게이트의 종류가 적게 되는 장점이 있다. <그림 10>의 코드는 클래스 Gate의 Intra_collapsing 프로세스의 의사 코드를 보여준다. Gate의 고장 표에서 중복된 고장을 찾아서 삭제한다. 여기서 고장 표는 <그림 8>의 CMOS 라이브러리의 Original Fault Table의 복사 본으로 각 게이트는 자신의 고장 표를 가진다.

4. 게이트간 중첩 프로세스(Inter_collapsing Process)

트랜지스터 합선 고장 모델에서 내부 중첩 프로세스 후에 게이트간에 등가 고장을 고려할 때 제안된 게이트간 중첩 프로세스의 흐름도를 <그림 11>에 보인다.

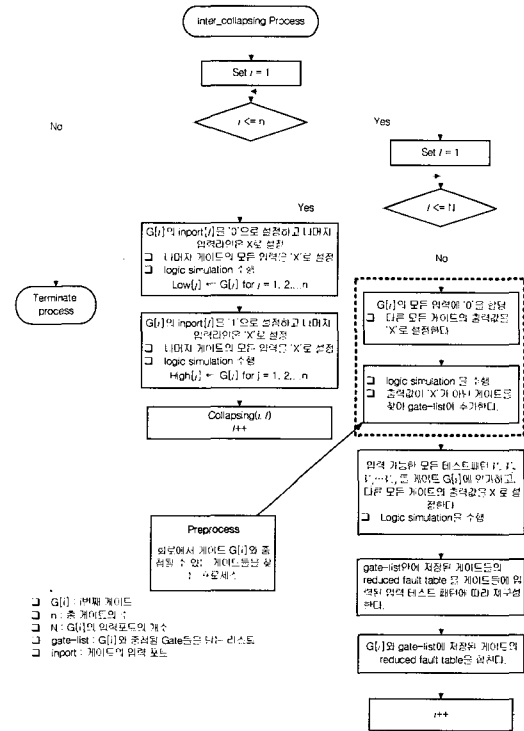


그림 11. 게이트간 중첩 프로세스의 흐름도
 Fig. 11. Flowchart of Inter_collapsing Process.

본 논문에서 고려하는 게이트간 중첩 프로세스는 다중 입력과 단일 입력 중첩을 단일화 된 루프를 통해서 검출 할 수 있고, 재구성 프로세스를 통해서 게이트간

더 많은 등가 고장의 상황을 만들어 줌으로 기존의 알고리즘에 비해서 효과적으로 게이트간에 등가 고장 중첩을 검출 할 수 있다. 또한 기존 방식과 다르게 게이트간 등가 고장을 검출하기 위해서 우선 하나의 게이트를 선택하는 방식에서 제안된 알고리즘에서는 선택된 게이트의 입력 단에 관련된 게이트들도 같이 추적하는 방식을 통해서 더욱 효과적인 게이트간 등가 고장 중첩을 수행한다.

게이트간 중첩 프로세스에서는 게이트의 단일 입력에 양의 값 고장과 음의 값 고장을 검출해내는 서브프로세스와 게이트의 전체 입력을 고려해서 게이트 간의 고장 표를 결합하는 서브프로세스로 구성이 된다. <그림 11>의 두 번째 판단문까지가 단일 입력의 양과 음의 값 고장을 검출하는 부분이다. 선택된 게이트 G[i]의 입력(import[i])을 하나 선택해서 입력에 '0'을 인가하고 다른 모든 입력과 G[i]가 아닌 게이트의 값에 'X'를 인

가하고 시뮬레이션을 수행한다. 시뮬레이션 수행 후 게이트 G[j](j = 1, 2, 3, ... n)의 출력 값을 Low[j]에 저장한다. 다음 단계로 선택된 입력에 '1'을 인가한 후 시뮬레이션을 수행한 후 각 게이트의 출력 값을 High[j]에 저장한다.

게이트 G[i]의 입력 포트 import[i]과 중첩될 수 있는 게이트를 검출하는 Collapsing 함수 동작은 <그림 12>와 같다. 우선 G[i]가 아닌 게이트 G[k]를 선택해서 G[k]의 입력에 연결된 게이트의 Low[m]과 High[m]을 검사한다. 만약 Low[m]과 High[m]이 'X'가 아니면서 서로 다른 값을 가지면, 게이트 G[i]의 입력 포트 import[i]의 양의 값과 음의 값 고장은 G[k]의 입력 포트의 양의 값과 음의 값 고장과 등가 되어 중첩이 된다.

다음 단계로 게이트 G[i]에 모든 가능한 테스트 패턴을 인가할 때 출력이 X가 아닌 게이트들을 검출하여 중첩을 수행하는 부프로세스이다. 게이트 G[i]에 하나의 테스트 패턴을 샘플로 인가하고 나머지 게이트의 입력을 'X'로 설정한 후 시뮬레이션을 수행하면 회로의 게이트의 출력 값 "X", "0", "1"을 갖는데, 이 중에서 출력 값이 X가 아닌 게이트는 G[i]와 중첩될 수 있는 게이트들이다. 본 알고리즘에서는 연산 횟수를 줄이기 위해 G[i]와 중첩될 수 있는 게이트들을 저장할 gate-list를 두어 G[i]에 샘플 테스트 패턴을 인가하였을 때 'X'가 아닌 게이트들을 저장한다.

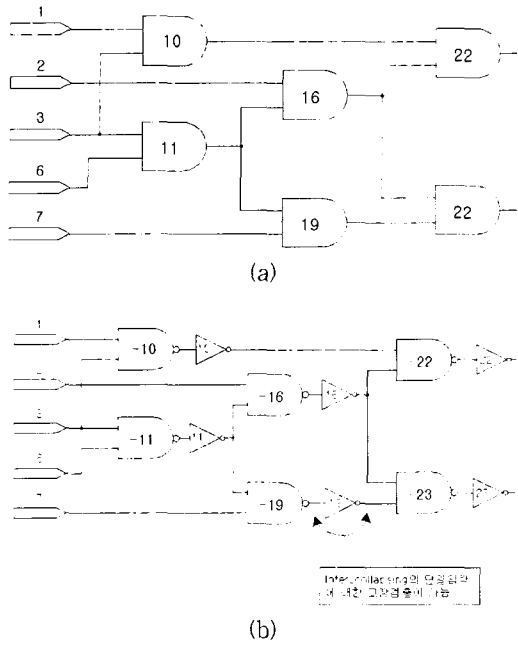
G[i]와 gate-list에 저장된 게이트들을 중첩하기 위해서 G[i]에 입력 가능한 패턴 $V_1, V_2, V_3, \dots, V_M$ 을 인가하고 나머지 게이트의 입력은 'X'로 설정한 후 시뮬레이션을 수행한다. G[i]의 고장 표와 중첩하기 위해 gate list에 저장된 게이트의 고장 표를 게이트의 입력 패턴에 따라 재구성한다. G[i]의 고장 표와 gate-list에 저장된 게이트의 고장 표와 중첩을 수행한다. 제안된 등가 고장 중첩 알고리즘과 기존의 알고리즘과의 비교 예를 <그림 13>에 보인다.

<그림 13(a)>회로는 6개의 AND 게이트로 이루어져 있으며 등가 고장 중첩하기 전의 전체 고장 수는 216개[AND 게이트의 고장 수 = $36 \times 6 = 216$]이다. 회로(a)는 게이트 내부 등가 고장 중첩을 거치면 각 AND 게이트의 고장 수가 12개로 줄어서 전체 고장은 72개까지 줄일 수 있다. 그러나 게이트간 등가 고장 중첩을 수행해도 등가 고장을 검출할 수 없다.

Function Collapsing (int i, int l)

- Step 1 : Set $k \leftarrow i+1$
- Step 2 : Set $j \leftarrow 1$
- Step 3 : G[k]의 입력 포트 import[j]에 연결된 게이트 G[m]을 찾는다.
- Step 4 : Low[m] = 0이고 High[m] = 1이면 G[k]의 입력 포트 import[j]와 G[i]의 import[i]은 등가로 중첩될 수 있다.
 즉, G[k]의 import[j]의 양의 값 고장과 G[i]의 import[i]의 양의 값 고장은 등가이고, G[k]의 import[j]의 음의 값 고장과 G[i]의 import[i]의 음의 값 고장은 등가이다.
- Step 5 : Low[m] = 1이고 High[m] = 0이면 G[k]의 import[j]의 양의 값 고장과 G[i]의 import[i]의 음의 값 고장은 등가이고, G[k]의 import[j]의 음의 값 고장과 G[i]의 import[i]의 양의 값 고장은 등가이다.
- Step 6 : $j \leftarrow j+1$
- Step 7 : $j \leq N$ 이면 Step 3으로 돌아간다(여기서 N은 G[k]의 입력 포트 수).
- Step 8 : $k \leq n$ 이면 Step 2로 돌아간다(여기서 n은 회로의 게이트 수).

그림 12. Collapsing 함수의 동작
 Fig. 12. Procedure of Collapsing function.



	고장 수(A)	게이트 내부(B)	게이트 내부 + 게이트간(C)	B/A	C/A
분해하기전 회로(a)	216	72	72	33.3%	33.3%
분해한 후 회로(b)	216	84	60	38.9%	27.8%

(c)

그림 13. 등가 고장 중첩 기법의 비교
Fig. 13. Comparison of equivalent fault collapsing methods.

제안된 방식의 알고리즘을 이용하는 (b) 회로는 회로 (a)의 AND 게이트를 NAND와 인버터로 분해할 경우, 전체 고장 수는 내부 중첩 프로세스를 거친 후 84개 [(NAND 고장 수 = 11 × 6 = 66) + (인버터 고장 수 = 3 × 6 = 18)]로 회로 (a)에 비해 12개가 증가했지만, 게이트간 중첩 프로세스를 거친 후 게이트의 단일 입력에서 양과 음의 값 고장을 검출해내는 서브프로세스의 12개와 게이트의 전체 입력을 고려해서 게이트 간의 고장 표를 결합하는 서브프로세스에서 12개[(NAND + 인버터)결합 = 고장 2개 감소] × 6 = 12개 감소를 검출하여 총 게이트간 24개의 등가 고장을 검출할 수 있다. 예를 들면 NAND (-23)게이트의 두 번째 입력 포트와 연결된 인버터 {19}의 입력의 양의 값 및 음의 값 고장이 등기이므로 두개의 등가 고장을 검출할 수 있다. 따라서 회로를 분해할 경우 전체 고장은 60개로

분해전보다 12개의 등가 고장을 더 검출할 수 있다.

트랜지스터 합선 고장 모델에서 제안된 등가 고장 중첩 알고리즘의 효율성을 검증하기 위해서 조합 회로로 구성되어 있는 ISCAS '85 벤치마크 회로에 적용한 모의 실험 결과를 <표 3>과 전체 고장 수를 고려할 경우, 고장 압축률에 관한 비교를 <그림 14>에 보인다. 모든 ISCAS '85 벤치마크 회로에서 기존의 제안된 방식보다 높은 등가 고장 중첩을 보이고 있다. <표 3>과 <그림 14>의 결과를 통해서 제안된 등가 고장 중첩 알고리즘이 IDDQ 테스트를 이용하는 트랜지스터 합선

표 3. 등가 고장 중첩을 이용한 실험 결과 비교

Table 3. Experimental result comparison of equivalent fault collapsing.

회로	게이트 수	고장 수 (A)	(B)	(C)	(D)	B/A(%)	C/A(%)	D/A(%)
C880	383	10188	3933	2889	2730	38.6	28.4	26.8
C1355	546	13080	5700	4006	2284	43.6	30.6	17.5
C1908	876	16788	6984	4845	4680	41.6	28.9	27.9
C2670	1193	27488	10258	7314	6986	37.3	26.6	25.4
C3540	1669	39672	14774	10194	9632	37.2	25.7	24.3
C5315	2307	60060	22880	16434	16180	38.1	27.4	26.9
C6288	2416	60672	26576	18523	12168	43.8	30.5	20.1
C7552	3512	79560	30480	21149	20008	38.3	26.6	25.2

(B) : 게이트 내부 고장 중첩 후 남은 고장 수, (C) : (게이트 내부 + 게이트간) 고장 중첩 후 남은 고장 수 (D) : 제안된 방식의 고장 중첩 후 남은 고장 수

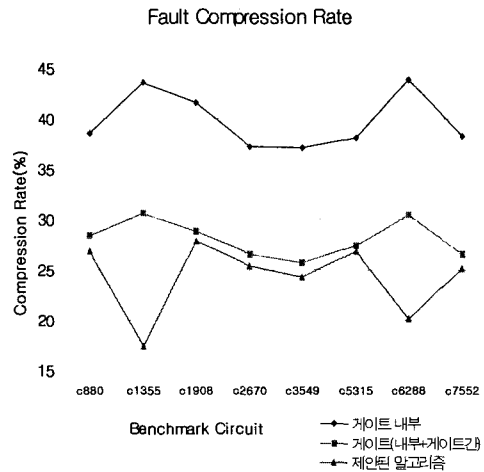


그림 14. 고장 압축률 비교
Fig. 14. Comparison of fault compression rate.

고장 모델에 더 효과적임을 알 수 있다. 이러한 결과는 트랜지스터 레벨 원리를 적용한 재조합 프로세서를 통한 분해와 다음 단계에서 이러한 성질을 고려하고 이용하는 Intra_collapsing과 Inter_collapsing 프로세서의 치환과정을 통해 등가 고장을 효율적으로 감소시킨 결과이다.

IV. 결 론

IDDQ 테스트는 CMOS VLSI 회로의 품질 및 신뢰성 향상에 중요한 테스트 방식이다. 그러나 상대적으로 느린 IDDQ 테스트를 위해서는 고려한 고장 모델에서 발생 가능한 고장의 수를 감소하거나 가능한 적은 수의 테스트 패턴을 유지하는 게 필요하다. 본 논문에서는 IDDQ 테스트 정보를 이용하여 테스트 대상 회로의 고장 검출, 고장 위치, 고장 진단에 효율적인 트랜지스터 합선 고장 모델을 위한 등가 고장 중첩 알고리즘을 제안하였다.

제안된 알고리즘은 트랜지스터 레벨 원리를 이용한 분해와 치환과정을 통해서 각 트랜지스터당 고려되는 고장의 수가 6개로 테스트 대상 회로의 게이트 수가 커질 경우, 고려 되어 할 고장의 수가 매우 크게 되는 트랜지스터 합선 고장 모델에 적용한 기법으로 ISCAS 벤치마크 회로를 통한 모의 실험을 수행하여 기존의 알고리즘에 비해서 효과적임을 확인하였다.

제안된 알고리즘을 IDDQ 테스트를 이용하는 트랜지스터 합선 고장 모델에 적용하여 Iddq용 테스트 패턴을 생성할 경우, 상대적으로 느린 전류 테스트의 단점을 보완할 수 있으며, 기존의 전압 테스트 방식에 비해서 CMOS VLSI 회로에 대한 높은 레벨의 신뢰성 있는 테스트가 가능하다.

참 고 문 헌

- [1] R. Rajsuman, *IDDQ Testing for CMOS VLSI*, Artech House, 1994.
- [2] 전병실 외, "기능테스트와 IDDQ 테스트를 위한 자체 점검 BIST 회로의 설계," 서울대학교 반도체 공학연구소 연구보고서, 1998
- [3] 홍성제 외, *테스팅 및 테스트를 고려한 설계*, 홍릉과학출판사, 1999
- [4] R. C. Aitken, "A Comparison of Defect Models for Fault Location with Iddq Measurements," *Proc. ITC'92*, pp. 778-787, Sept. 1992.
- [5] 전병실 외, "합선고장을 위한 IDDQ 테스트 패턴 생성에 관한 연구," *한국통신학회논문지*, vol. 25, no. 12-A, pp. 1904-1911, 2000
- [6] W. Mao and R. K. Gulati, "QUIETEST: A Quiescent Current Testing Methodology for Detecting Leakage Faults," *Proc. ICCAD'90*, pp. 280-283, 1990.
- [7] S. Chakravarty and S. T. Zachariah, "STBM: A Fast Algorithm to Simulate IDDQ Tests for Leakage Faults," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 5, pp. 568-576, May 2000.
- [8] X. Wen, H. Tamamoto and K. Kinoshita, "IDDQ Test Vector Selection for Transistor Short Fault Testing," *System and Computers in Japan*, vol. 28, no. 5, 1997.
- [9] X. Wen, H. Tamamoto and K. Kinoshita, "Transistor Leakage Fault Location with IDDQ Measurement," *Proc. ATS'95*, pp. 195-200, Nov. 1995.

저 자 소 개

裴 晟 桓(正會員) 第10卷 TC編 第9號 參照