

論文2000-37SD-11-13

# 순차 회로의 효율적인 지연 고장 검출을 위한 새로운 테스트 알고리즘 및 스캔 구조

## (Efficient Delay Test Algorithm for Sequential Circuits with a New Scan Design)

許 璟 會\*, 康 容 碩\*, 姜 成 昊\*

(Kyung-Hoi Huh, Yong-Seok Kang, and Sungho Kang)

## 요 약

지연 고장을 위한 테스트는 디지털 회로의 속도와 직접도가 크게 향상되면서 필수적인 것으로 생각되고 있다. 그러나, 순차 회로에는 상태 레지스터들이 있기 때문에, 지연 고장을 검출하는 것이 쉽지 않다. 이러한 난점을 해결하기 위해 회로의 단일 고착 고장과 지연 고장을 효율적으로 검출할 수 있는 새로운 테스트 방법과 알고리즘을 개발하였고, 이를 적용하기 위한 새로운 구조의 스캔 플립-플롭을 제안한다. ISCAS 89 벤치마크 회로에 대한 실험을 통해 지연 고장 검출률이 기존의 전통적인 스캔 테스트 방법에 비해 현격하게 향상된 것을 알 수 있다.

## Abstract

Delay testing is essential for assurance of digital circuits as the speed and the density of the circuits improve greatly. However, delay faults in sequential circuits cannot be detected easily due to the existence of state registers. To overcome this difficulty a new test method and algorithm are devised which can be used for both stuck-at testing and delay testing. To apply the new test method, a new scan flip-flop is implemented. Experimental results on ISCAS 89 benchmark circuits show that the number of testable paths can be increased drastically over conventional scan techniques.

## I. 서 론

디지털 회로가 빨라지고 집적되면서, 회로를 좀 더 효율적으로 테스트하는 것이 중요하게 되었다. 게다가, 고성능의 칩 개발에 대한 요구가 증가하면서 회로의 기능뿐만 아니라, 신호 천이 지연을 검사하는 것이 필요하게 되었다. 따라서, 제조 공정에서 생긴 결함을 검출하기 위한 고착 고장과 지연 고장의 테스트가 많이

사용되는데, 지연 고장 테스트의 목적<sup>[1]</sup>은 생산된 회로가 요구되는 클럭 주파수 내에서 제대로 동작하는지를 확인하는 것이다.

시스템의 시간에 관련된 결함은 집중된 지연 고장으로 인해 생기거나 경로에 흩어진 분산된 지연 고장들이 합쳐져서 생기게 된다. 두 종류의 지연 고장 모델이 개발되어 널리 쓰이고 있는데 이는 게이트 지연 고장 모델(gate delay fault model)<sup>[2,3]</sup>과 경로 지연 고장 모델(path delay fault model)<sup>[4,5]</sup>이다. 게이트 지연 고장 모델은 지연이 회로 내의 게이트 입력이나 출력에 집중되었다고 가정하는 것이므로 회로내의 각 게이트에 널리 퍼져 있는 작은 지연으로 생긴 결함은 검출 불가능하다. 반면에, 경로 지연 고장 모델은 시험 중인 경로를 따라 있는 작은 지연들이 모여서 회로가 오동작 하도록 하는 것을 말한다. 따라서, 이 모델에

\* 正會員, 延世大學校 電氣工學科

(Dept. of Electrical Eng., Yonsei Univ.)

※ 본 연구는 1997년도 한국학술진흥재단 대학부설연구 소과제 연구비에 의하여 연구되었음.

接受日字:2000年 3月30日, 수정완료일:2000年 10月31日

기초를 둔 테스트는 회로의 경로에 따른 집중 또는 분산된 결함을 모두 검출할 수 있다. 또, 회로 내의 긴 경로들의 고장에 대해 클럭 속도에 따른 테스트 성공 여부에 관한 정확한 모습을 제공하므로 속도에 따라 칩을 분류하는 데 쓰일 수 있다. 그러나, 회로 내의 모든 경로를 테스트하는 것은 사실상 불가능한데, 그것은 경로의 수가 회로의 게이트 수에 지수함수적으로 증가하기 때문이다.

경로 지연 고장에 대한 테스트는 경로의 시작점에서 천이를 발생시킬 수 있도록 하는 두 개의 벡터로 이루어진다. 발생된 천이는 경로를 따라 경로의 끝까지 전파되어 관찰 가능하게 된다. 이 때, 경로를 따라 전파되는 천이가 경로의 끝까지 전파되는데 걸리는 시간이 해당 경로의 지연 시간이 된다. 이러한 경로 지연에 대한 테스트를 위한 많은 연구가 지금까지 진행되어 오고 있다<sup>[5-7]</sup>.

경로 지연 고장에 대한 테스트에는 크게 무해저드 경성(hazard free robust(HFR)) 테스트, 경성(robust(ROB)) 테스트, 연성(non-robust(NR)) 테스트의 세 종류로 나눌 수 있다<sup>[8]</sup>. 무해저드 경성 테스트는 경로에의 신호들에 동적 해저드(dynamic hazard)가 없고 회로의 다른 부분에서의 지연과 무관하게 경로상의 지연 고장을 검출할 수 있는 두 패턴의 테스트이다. 경성 테스트는 회로의 다른 부분에서의 지연과 무관하게 경로상의 지연 고장을 검출할 수 있는 두 패턴의 테스트로 경로에 특정 천이를 유발시켜야 하고 경로상의 모든 신호들은 특정 천이가 도달하기 전까지 최종 값을 알 수 없어야 한다. 경성 테스트는 시험중인 경로가 초과 지연을 가지면 실패하지만, 회로 내의 다른 경로가 초과 지연을 갖는 경우에도 실패한다. 따라서 회로에 대한 테스트의 목적이 진단이 아니고 단지 고장 검출 만이라면 경성 테스트를 구하는 것이 무해저드 경성 테스트를 구하는 것보다 유리하다. 경성 테스트의 요건을 갖추지 못한 테스트를 연성 테스트라고 부른다. 이는 두 개로 다시 구분될 수 있다. 먼저 경성 테스트에 가까운 연성 테스트를 강연성(strong non-robust(SNR)) 테스트라 부른다. 강연성 테스트는 모든 요건이 경성 테스트와 같지만 정적 해저드(static hazard)가 있어도 무방한 두 패턴의 테스트이다<sup>[8]</sup>. Schulz<sup>[9]</sup>는 두 번째 사이클의 경우에 경성 테스트와 요구되어지는 값이 같고 첫 번째 사이클의 경우 요구되어지는 것이 경로의 시작 부분의 초기값으로만

결정되는 방법으로 연성 테스트를 구하였는데 이를 약연성(weak non-robust(WNR)) 테스트라고 부른다. 약연성 테스트를 이용할 경우는 경로를 제외한 회로의 나머지 부분에서는 지연 고장이 없다는 보장이 있어야 경로상의 지연 고장을 검출할 수 있다는 단점이 있다.

이 네 가지 종류의 테스트를 정의에 따라 비교하면 약연성, 강연성, 경성, 무해저드 경성 순으로 테스트를 구하기 위한 요건이 많아진다. 요건이 많아질수록 테스트하기 힘들고, 따라서 위의 네 가지 종류에서 무해저드 경성 테스트가 가장 양질의 테스트이다.

조합 회로에 있어서 어떤 고장 경로의 천이 지연을 검출하기 위해서는 입력 래치(latch)와 출력 래치에 그림 1<sup>[4,10]</sup>과 같이 두 단계의 클럭을 가해야 한다. 첫 번째 단계에서, 초기화 벡터( $V_1$ )가 입력 래치에 저장된다. 초기화 벡터가 안정화된 후 천이 벡터( $V_2$ )가 시간  $t_0$ 에서 입력 클럭에 의해 입력 래치에 가해진 후, 시스템 클럭만큼의 시간 후에 출력 클럭이 가해지면, 출력 래치에 출력값이 저장되게 된다. 이 때, 저장된 값이 고장이 존재하지 않는 회로와 다르다면 테스트를 수행한 회로에 고장이 존재한다는 것을 의미하므로 검출 가능하다.

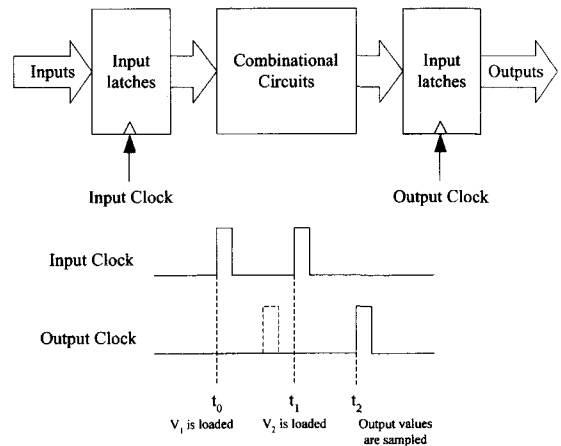


그림 1. 지연 고장 테스트를 위한 하드웨어 모델

Fig. 1. Hardware model for delay fault testing.

순차 회로에서는 조합 회로와는 다르게 지연 고장을 위한 테스트를 생성하는 것이 힘들다. 그것은 순차 회로의 출력이 주입력뿐만 아니라 초기 상태에 따라서도 달라지기 때문이다. 즉, 초기 상태를 조절하는 방법만으로는 테스트에 필요한 모든 벡터를 얻을 수 없다는

것이고, 따라서 고장 검출률도 매우 낮게 된다. 이러한 문제를 해결하기 위해 기억 소자에 초기 상태를 저장할 수 있는 스캔 플립-플롭이 사용된다. 이러한 방법은 조절 용이도와 관측 용이도를 높임으로써 고착 고장에 대한 테스트를 수월히 수행할 수 있는 장점이 있다. 그러나, 스캔 플립-플롭의 사용만으로는 두 개의 벡터를 모두 조절할 수 없으므로, 지연 고장의 검출률을 원하는 수준으로 높일 수는 없다.

일반적으로, 표준 스캔(standard scan)을 사용하는 지연 고장 테스트 방법으로는 스캔 이동(scan shifting) 방법<sup>[1]</sup>과 기능적 지정(functional justification) 방법<sup>[9,11]</sup>이 있다. 스캔 이동 방법은 앞단의 스캔에 저장된 첫 번째 테스트 논리 값을 바로 뒷단의 스캔으로 이동하여 두 번째 테스트 논리 값으로 사용하는 방법이다. 따라서 이러한 방법의 경우에는 테스트 가능한 많은 경로에 대해 테스트를 생성하지 못할 수가 있다.

이러한 주사 이동의 문제점을 해결하고 보다 정확한 지연 고장 검사를 위해 기능적 지정 방법이 사용된다. 이 방법에서 첫 벡터는 스캔 플립-플롭으로 입력되어 들어가고, 두 번째 벡터는 회로의 기능에 따라 정해진다. 그러나, 이 역시 두 벡터가 두 시간대 동안 테스트에 필요한 논리 값들을 만족시켜야 한다는 문제점을 가지고 있다. 게다가, 첫 번째 시간대에 필요한 테스트 패턴들의 논리 값들은 'X(don't care)'를 많이 갖는다. 따라서 이러한 방법은 스캔 플립-플롭의 높은 조절 용이도를 임의의 값으로 채우는 것으로 낭비할 수 있다. 이는 각 소자 유형에 따른 경로의 입력 조건을 보여주고 있는 표 1을 통해 알 수 있다.

표 1. 경로 지연 테스트의 경로의 입력 조건  
Table 1. Off-path input constraints of path delay test.

소자 유형	경로 입력 천이	경로의 입력 천이			
		HFR	ROB	SNR	WNR
AND /NAND	상승 천이	11	X1	X1	X1
	하강 천이	11	11	11	X1
OR /NOR	상승 천이	00	00	00	X0
	하강 천이	00	X0	X0	X0

확장 스캔(enhanced scan)은 두 개의 패턴을 저장하기 위해 두 개의 플립-플롭을 사용하므로, 순차 회로를 조합 회로화할 수 있다. 그러나, 추가되는 플립-

플롭만큼의 면적 오버헤드가 증가하므로 실질적인 순차 회로에는 거의 사용되지 못하고 있다.

따라서, 순차 회로의 지연 고장에 관한 대부분의 연구들은 스캔을 사용하는 환경에서 면적 오버헤드를 줄이면서 고장 검출률을 높이는 방법에 관해 관심을 두고 있다. 이러한 관점에서 몇몇 연구가 시도되어져 왔다. Savir<sup>[12]</sup>는 효율적인 지연 고장 검출을 위해 새로운 구조의 스캔 래치를 설계한 바가 있다. 그러나 순차 회로에 적용하기에 충분한 연구는 아직 이루어지지 못하고 있다.

기존의 지연 고장 테스트 방법들을 극복하기 위해 본 논문에서는 효율적인 지연 고장 테스트 알고리즘을 제시한다. 다음 장에서는 새로운 테스트 방법과 그에 따른 ATPG 도구의 적용 방법을 설명하고, III장에서는 II장에서 설명한 테스트 방법을 실제 적용하기 위한 새로운 스캔 구조를 제안한다. 새로운 테스트 알고리즘에 따른 테스트 순서는 IV장에서 다루어지며, V장에서는 새로운 테스트 알고리즘을 적용한 결과를 보이고 이에 대한 의미를 설명한다. 마지막 VI장은 결론으로써, 본 연구의 내용을 요약하고, 새로운 과제를 제시한다.

## II. 스캔 지정을 이용한 테스트 방법

### 1. 테스트 방법

테스트 방법을 좀 더 명확히 설명하기 위해, 우선 몇 가지 용어들을 정리하면 다음과 같다.  $P1$ ,  $P2$ 를 각각 첫 번째 테스트 패턴과 두 번째 테스트 패턴이라고 생각한다. 좀 더 세부적으로  $P1$ 은  $Pp1$ 과  $Ps1$ 로 구성되는데,  $Pp1$ 은 첫 번째 테스트 패턴 중 주입력에 가해지는 패턴을 나타내고,  $Ps1$ 은 플립-플롭에서 가해지는 패턴을 의미한다.  $P1$ 과 마찬가지로  $P2$ 도  $Pp2$ 와  $Ps2$ 로 구성되며, 의미하는 바는  $Pp1$ ,  $Ps1$ 과 동일하다. 그리고 마지막으로  $Ps1$ 을 생성할 수 있는 주입력 패턴을  $Pp0$ 로, 스캔 입력을  $Ps0$ 로 표시한다. 출력의 경우에는,  $R1$ 과  $R2$ 를  $P1$ 과  $P2$ 의 응답 패턴을 의미하는 것으로 각각 나타낸다.

앞서 언급했듯이, 기능적 지정 방법은 조절용이도가 높은 스캔 플립-플롭을 'X' 값이 많은 첫 번째 테스트 패턴을 저장하는 데 사용한다. 이러한 낭비를 없애기 위해, 두 번째 테스트 패턴을 스캔 플립-플롭에 저장하는 방법을 제안한다. 반대로, 첫 번째 테스트 패턴은

회로의 기능에 의해 지정되도록 할 수 있고, 따라서 이러한 방법을 스캔 지정(scan justification) 방법이라고 정의한다. 이러한 방법에 따르면, 첫 번째 테스트 패턴이 회로의 입력에 가해지기 전에 두 번째 테스트 패턴이 회로의 스캔 안에 저장되어 있어야 한다. 두 번째 테스트 패턴을 저장한 상태에서 첫 번째 테스트 패턴을 같은 경로로 통과시키는 것은 불가능하기 때문에, 추가적인 래치가 필요하다. 게다가 스캔 지정 방법을 적용하기 위해서는 *PsI*을 회로의 기능을 통해 효율적으로 생성하는 방법이 필요하다. 본 논문에서는 주입력과 스캔 플립-플롭을 모두 조절하는 방법을 제안한다.

2. ATPG를 이용하여 *PsI*을 생성하는 방법

실제적인 순차 회로에 스캔 지정 방법을 적용하기 위해서, 본 논문에서는 ATPG를 이용하여 *PsI*을 생성하는 방법을 제시한다. ATPG 도구는 주어진 고착 고장에 대한 테스트 패턴을 생성하는 프로그램이다. 순차 회로에서 플립-플롭을 모두 제거한 조합 회로 부분을 고려했을 때, *PsI* 패턴 중 '0' 값을 필요로 하는 출력단에 모두 인버터를 삽입하면, 조합 회로 부분의 출력은 '1' 값이나 'X' 값을 가지게 될 것이다. 이제, 주출력 부분을 제외한, 나머지 플립-플롭과 연결된 출력들 중에 '1' 값을 갖는 것들을 모두 하나의 AND 게이트의 입력으로 묶고, 이 AND 게이트의 출력 부분에 0-고착 고장을 삽입한다. ATPG는 이 고장에 대한 테스트 패턴을 생성할 것이고, 생성된 패턴들이 바로 *PsI*을 만들어 낼 수 있는 입력 패턴이다.

그림 2는 S27 순차 벤치마크 회로의 경우에 수정된 조합 회로 부분을 나타낸다. 예를 들어 만약, 필요한 첫 번째 테스트 패턴이 '10X(G11, G10, G13)'라고 하면, G11 게이트의 출력은 AND 게이트의 입력과 직접 연결시킨다. '0' 논리 값을 갖는 G10 게이트와 AND 게이트 사이에는 인버터를 삽입하고, 'X' 논리 값을 갖는 G13은 수정을 할 필요가 없다. 그림 2에서 첫 번째 테스트 패턴에 따라 추가된 게이트들을 회색으로 칠하여 표시하였다. AND 게이트의 출력부에 0-고착 고장을 삽입한 후, ATPG 도구를 시행한다. ATPG 도구는 결과로서 *PsI*을 생성할 수 있는 *Pp0*과 *Ps0*을 제공해 준다.

ATPG의 고장 검출률은 스캔 지정 방법의 성능을 직접적으로 표시하는 중요한 의미가 있는 값이다. 왜냐하면, 만약 ATPG 도구가 *P0*을 생성하지 못한다면 그에 해당하는 경로에 대한 지연 고장은 테스트할 수

없기 때문이다.

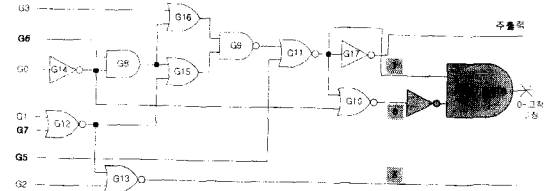


그림 2. S27 순차 벤치마크 회로의 수정된 조합 회로 부분

Fig. 2. Modified combinational part of S27 benchmark circuit.

III. 새로운 스캔 플립-플롭 설계

스캔 지정 방법을 적용할 때의 요구 조건을 만족시키기 위해서 새로운 스캔 플립-플롭을 고안해 내었다. 이 스캔 플립-플롭은 *Ps2*를 저장한 채로 *PsI*을 통과시킬 수 있어야 하기 때문에, 면적 오버헤드를 감수해야만 하지만, 기능적 지정 방법과 스캔 지정 방법을 혼용할 수 있으므로 지연 고장 검출률을 많이 높일 수 있다.

그림 3은 새로운 스캔 플립-플롭 구조를 CMOS 수준으로 설계한 것이다. 이 스캔 플립-플롭은 세 개의 래치(L1, L2, L3)와 두 개의 멀티플렉서로 구성되어 있는데, 회색으로 칠한 소자들은 표준 스캔 플립-플롭과 비교해 추가된 것들이다. 실제 회로에서는 *Data in*과 *Scan in*, 그리고 스캔의 출력에 인버터가 붙는데, 각각의 새로운 스캔 플립-플롭마다의 면적 오버헤드는 이를 포함하여 50퍼센트 정도이다. 이러한

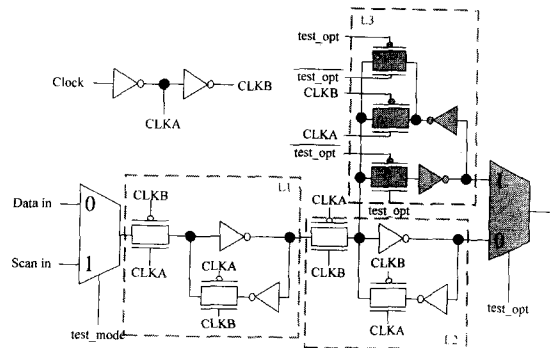


그림 3. 새로운 스캔 플립-플롭  
Fig. 3. New scan flip-flop.

오버헤드는 문제가 될 수 있지만, 지연 고장 검출률을 확장 스캔의 수준으로 향상시킬 수 있다면 이 정도의 오버헤드는 충분히 받아들일 수 있다. 게다가 확장 스캔 플립-플롭의 오버헤드가 약 80퍼센트에 이르는 것을 고려한다면 더욱 고려해볼 만 하다고 할 수 있다. *test\_opt*은 외부 핀으로 추가되어 어떤 테스트 방법으로 테스트를 할 것인지를 선택할 수 있게 한다.

정상 동작 모드(*test\_mode*와 *test\_opt* 신호가 모두 '0' 값을 가질 때)에서, 이 스캔 플립-플롭은 보통의 플립-플롭과 똑같이 동작한다. 스캔 이동 모드로는 다음과 같은 두 가지가 있는데, 첫 번째는 *test\_mode*가 '1' 값을, *test\_opt*이 '0' 값을 가질 때로서, 두 번째 래치(L2)를 슬레이브로 사용하여 스캔 이동을 수행하는 모드이다. 반면에, *test\_mode*와 *test\_opt* 모두 '1' 값을 가졌을 때는 세 번째 래치(L3)를 슬레이브로 사용하여 스캔 이동을 수행한다. 기능적 지정 방법 역시 이 스캔 플립-플롭을 사용하여 적용할 수 있다. 만약, *test\_opt* 신호가 항상 논리 '0'으로 조정된다면, 표준 스캔 플립-플롭과 마지막에 추가된 멀티플렉서에서의 지연 시간외에는 다를 것이 전혀 없다.

마지막으로 특별한 모드가 추가되었는데, 이를 클럭 모드라고 부르기로 한다. 즉, *test\_mode*는 논리 '0'의 값을 그리고 *test\_opt*은 '1' 값을 가질 때인데, 이 때는 *test\_opt* 신호가 보통의 클럭 신호와 같은 역할을 한다. 이에 관해서는 IV장에서 좀 더 자세히 다루기로 한다. 표 2는 이와 같은 모드들을 제어 신호인 *test\_mode*와 *test\_opt*에 따라 구분지어 보여준다.

표 2. 동작 모드의 종류  
Table 2. Types of operational modes.

<i>test_mode</i>	<i>test_opt</i>	동작 모드
0	0	정상 동작 모드
0	1	클럭 모드
1	0	L2-스캔 이동 모드
1	1	L3-스캔 이동 모드

#### IV. 테스트 알고리즘

순차 회로에서 지연 고장의 테스트를 행하는 방법은 기능적 지정 방법과 스캔 지정 방법에 따라 차이가 난다. 기능적 지정 방법을 사용할 경우에는, *test\_opt* 신호가 항상 '0' 값으로 고정된 상태에서 우선 L2에 첫 번째 테스트 패턴을 가져다 놓는다. 이제, 입력 값이 주출력까지 뻗어나갈 수 있을 만큼의 충분한 시간을 줄 수 있는 느린 클럭을 가하고, 연속적으로 보통 클럭을 한 주기만큼 가한다. 결과적으로, 이에 대한 응답 패턴이 스캔 플립-플롭에 저장되고, 이들은 스캔 출력을 통하여 모두 관찰 가능하다.

스캔 지정 방법의 개요는 그림 4에 나타나 있다. 이 그림에서 회색으로 칠해진 화살표는 필요한 데이터의 흐름을 나타낸다. 처음에, *Ps2*가 L3-스캔 이동 모드를 통해 L3에 저장되고, 다시 *Ps0*가 L2-스캔 이동 모드를 통해 L2에 이동되어 들어간다. 첫 번째 테스트 패턴을 생성하기 위해, *Pp0*가 주입력에 준비되고, 정상 동작 모드에서 느린 클럭이 가해지면, *Ps1*이 L1에 도달하게 된다. 느린 클럭이 한 주기 더 가해지는 동시에 주입력단에는 *Pp1*이 활성화되면 회로는 초기화 되게 되고, 이제 두 번째 테스트 패턴을 가하기만 하면 된다. *Ps2*가 L3에 저장되어 있기 때문에, *test\_opt* 신호는 '1' 값을 가져야 한다. 그러나, 만약 *test\_opt* 신호와 클럭 신호가 동시에 '1' 값을 갖게 된다면 L3은 마지막 테스트 과정 동안 *Ps2*를 유지할 수 없게 된다. 따라서 본래의 클럭 신호는 '0' 값으로 유지되고, 대신에 *test\_opt*이 보통 클럭과 같은 역할을 하게된다. 이를 위해, *test\_opt* 신호가 '1' 값을 유지하는 시간은 보통 클럭 주기와 정확히 같아야 한다. 게다가, L1은 *test\_opt* 신호가 다시 '0' 값으로 돌아오는 동시에 포착 모드로 가야하는데, 이것은 원하지 않는 출력값에 의해 L1에 저장된 *Rs2* 값이 영향을

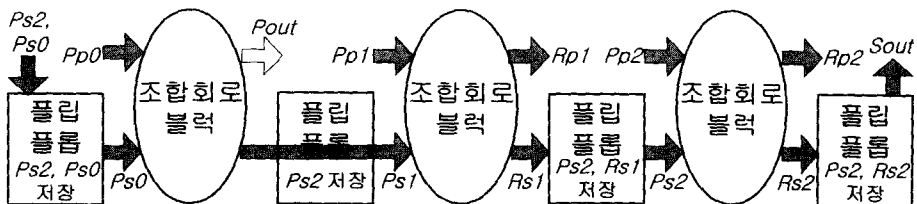


그림 4. 스캔 지정 방법의 개요  
Fig. 4. Scheme of the scan justification method.

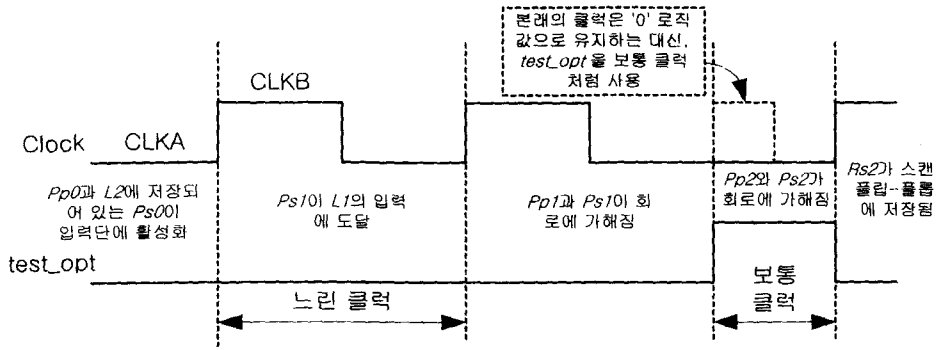


그림 5. 테스트 순서  
Fig. 5. Timing diagram of testing.

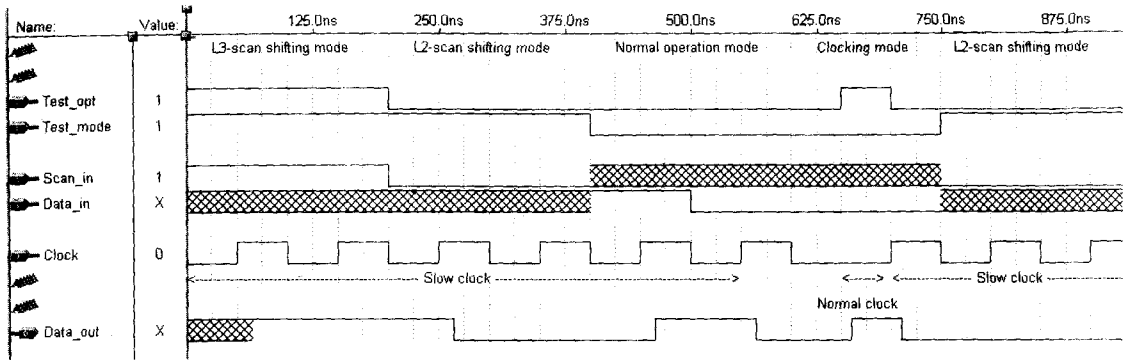


그림 6. 테스트 과정의 시뮬레이션  
Fig. 6. Timing simulation of testing.

받지 않게 하기 위해서이다. 이러한 이유 때문에, *test\_opt*이 논리 '0'으로 돌아올 때, 클럭은 '1' 값을 갖도록 되어야 한다. 결과적으로 이러한 테스트 과정 후에 *Ps2*는 스캔 플립-플롭에 저장되어 있으므로 스캔 출력을 확인하여 신호 전이가 일어났는지를 확인할 수 있다. 시간에 따른 전체적인 테스트 순서는 그림 5에 나타나 있다.

그림 6은 전체적인 테스트 과정의 예를 시뮬레이션을 통해 나타낸 것이다. 이 그림에서는 클럭 모드에 관해 좀 더 주의 깊게 살펴볼 필요가 있다. 처음에, '1' 값이 L3-스캔 이동 모드를 통해 L3에 이동되어 들어가고 *test\_opt*신호가 다시 '1'로 될 때까지 그 값을 저장하게 된다. 정상 동작 모드에서, *Data\_out*은 *Data\_in* 신호를 따라가다가 클럭 모드로 바뀌면서 *Data\_in* 신호가 나타내는 '0'을 따라가지 않고 '1'로 바뀌는 것을 볼 수 있다. 이것은 *test\_opt*가 '1'로 바뀌면서 L3에 저장되어 있던 '1' 값이 *Data\_out*으로 나가게 되기 때문이다. 즉, 클럭을 사용하지 않고,

*test\_opt* 신호를 이용하여, 두 번째 테스트 패턴 값을 출력으로 내보내게 되며, 이런 식으로 새로 제안된 스캔 플립-플롭은 스캔 지정 방법을 적용할 수 있다.

효과적인 테스트를 수행하기 위해 위의 두 방법, 즉 기능적 지정 방법과 역기능적 지정 방법을 같이 사용함으로써 새로운 테스트 알고리즘을 개발할 수 있다. 두 가지 방법을 동시에 이용하므로 새로운 테스트 알고리즘을 이용하면 지연 고장을 위한 테스트 패턴을 보다 많은 경로에 대해 생성해 낼 수 있다. 그림 7은 이러한 알고리즘을 보여준다. 우선 스캔 지정 방법을 수행하기 위해, 조합 회로 부분에 대한 가능한 모든 지연 고장 테스트를 생성한다. 구해진 테스트 패턴 중에 어떠한 패턴들을 회로의 기능에 의해 지정할 수 있는지에 대해 2장에서 살펴본 방법으로 조사하며, 이때 P0 역시 얻게된다. 여기서 중요한 것은 어떠한 경로들이 테스트 가능한 지, 가능하지 못한 지에 대한 정보를 저장하여, 기능적 지정 방법을 적용할 때 테스트되지 못한 경로에 대해서만 조사할 수 있도록 해

```

/*----- Scan justification -----*/
{
Generate delay tests for the combinational logic(Get the test
vectors P1, P2);
for (i = 0; i < Num_tests; i++) /* Num_tests : number of delay
tests for the combinational logic */
{
Modify the combinational logic corresponding to the each
first test pattern(Ps1(i));
ATPG for the inserted stuck-at-0 fault;
if (ATPG is succeeded)
{
Memorize the input pattern(a part of P0) and the tested
path(a part of TP);
Num_success++;
}
}
for (i = 0; i < Num_success; i++)
P0, P1, P2 are applied to the sequential circuit;
}

/*----- Functional justification -----*/
{
Generate delay tests using functional justification:
Compare TP with the tested paths;
if (there exist the paths which can be only detected by
functional justification)
{
for (i = 0; i < Num_paths; i++) /* Num_paths : the number
of paths which can be only detected by functional
justification */
{
Apply the test corresponding to the each path(Path(i))
to the sequential circuit.
}
}
}
    
```

그림 7. 테스트 알고리즘  
Fig. 7. Test algorithm.

야 테스트 시간을 절약할 수 있다. 이제 얻어진 세 개의 테스트 패턴, P0, P1, P2를 순차적으로 회로에 가해 테스트를 수행할 수 있다.

기능적 지정 과정은 선택적이다. 즉, 만약 위의 스캔 지정 방법을 수행한 후 원하는 고장 검출률을 얻지 못하였다면 이 과정을 추가할 수 있다. 기능적 지정 과정은 기능적 지정 방법을 적용하여 테스트를 생성하는 것으로 시작된다. 이 방법으로 테스트 가능한 경로들 중에 만약 스캔 지정 과정에서 테스트 불가능한 경로가 있다면 해당하는 테스트 패턴을 회로에 가해 테스트를 수행할 수 있다. 이것은 스캔 지정 과정에서 테스트 가능한 경로들에 대해 미리 기억해 두었기 때문에 가능하다.

이러한 새로운 테스트 알고리즘은 두 개의 과정으로 이루어져 있기 때문에, 테스트 시간과 고장 검출률간의 타협점을 쉽게 결정할 수 있는 장점을 가지고 있다. 만약, 회로를 빠르게 테스트하고 싶다면 기능적 지정 방법만을 적용할 수도 있다. 그렇지 않고 좀 더 정확한 테스트를 원한다면, 전체적인 테스트 알고리즘을 통해 회로의 정확한 동작을 보장할 수 있을 것이다.

V. 결 과

표 3, 4, 5, 6은 ISCAS'89 벤치마크 회로들에 대해 경로 지연 고장 테스트를 수행한 결과이다. 괄호 안의 두 수는 전체 경로 수에 대한 테스트 가능한 경로 수를 의미한다. 5000개 미만의 경로를 갖는 회로에

대해서는 모든 경로에 대해 고려하였고, 5000개 이상의 회로들에 대해서는 5000개의 임의의 경로들에 대해서 실험하였다.

표 3. 테스트 결과(무해저드 경성)  
Table 3. Result of test.(hazard free robust)

벤치마크 회로	기능적 지정 [%] (테스트 가능한 경로 수/전체 경로)	확장 스캔 [%] (테스트 가능한 경로 수/전체 경로)	새로운 테스트 알고리즘 [%] (테스트 가능한 경로 수/전체 경로)
S838	3.07(62/2018)	100.00(2018/2018)	59.22(1195/2018)
S953	32.00(740/2312)	99.13(2292/2312)	84.13(1945/2312)
S1423	38.60(1930/5000)	93.52(4676/5000)	89.26(4463/5000)
S1494	25.08(1254/5000)	99.08(4954/5000)	89.48(4474/5000)
S9234	44.02(2201/5000)	78.76(3938/5000)	77.58(3879/5000)
S13207	10.44(522/5000)	22.84(1142/5000)	18.04(902/5000)
S15850	33.86(1693/5000)	79.76(3988/5000)	75.98(3799/5000)

표 4. 테스트 결과(경성)  
Table 4. Result of test.(Robust)

벤치마크 회로	기능적 지정 [%] (테스트 가능한 경로 수/전체 경로)	확장 스캔 [%] (테스트 가능한 경로 수/전체 경로)	새로운 테스트 알고리즘 [%] (테스트 가능한 경로 수/전체 경로)
S838	31.47(635/2018)	100.00(2018/2018)	81.91(1653/2018)
S953	41.22(953/2312)	99.57(2302/2312)	95.50(2208/2312)
S1423	41.80(2090/5000)	95.86(4793/5000)	94.10(4705/5000)
S1494	41.12(2056/5000)	99.52(4976/5000)	96.84(4842/5000)
S9234	47.08(2354/5000)	81.84(4092/5000)	81.16(4058/5000)
S13207	12.70(635/5000)	27.48(1374/5000)	21.42(1071/5000)
S15850	37.36(1868/5000)	82.72(4136/5000)	80.98(4049/5000)

표 5. 테스트 결과(강연성)  
Table 5. Result of test.(Strong non-robust)

벤치마크 회로	가능적 지정 [%] (테스트 가능한 경로 수/전체 경로)	확장 스캔 [%] (테스트 가능한 경로 수/전체 경로)	새로운 테스트 알고리즘 [%] (테스트 가능한 경로 수/전체 경로)
S838	31.47(635/2018)	100.00 (2018/2018)	81.91(1653/2018)
S953	41.57(961/2312)	100.00(2312/2312)	96.45(2230/2312)
S1423	41.80(2090/5000)	96.46(4823/5000)	94.10(4705/5000)
S1494	53.82(2691/5000)	99.78(4989/5000)	96.16(4808/5000)
S9234	47.94(2397/5000)	88.02(4401/5000)	87.00(4350/5000)
S13207	14.16(708/5000)	29.84(1492/5000)	24.08(1204/5000)
S15850	37.64(1882/5000)	86.70(4335/5000)	84.62(4231/5000)

표 6. 테스트 결과(약연성)  
Table 6. Result of test.(weak non-robust)

벤치마크 회로	가능적 지정 [%] (테스트 가능한 경로 수/전체 경로)	확장 스캔 [%] (테스트 가능한 경로 수/전체 경로)	새로운 테스트 알고리즘 [%] (테스트 가능한 경로 수/전체 경로)
S838	56.59(1142/2018)	100.00(2018/2018)	100.00(2018/2018)
S953	66.61(1540/2312)	100.00(2312/2312)	100.00(2312/2312)
S1423	64.00(3200/5000)	97.10(4855/5000)	97.14(4857/5000)
S1494	83.84(4192/5000)	99.78(4989/5000)	99.78(4989/5000)
S9234	64.94(3247/5000)	91.72(4586/5000)	91.70(4585/5000)
S13207	19.54(977/5000)	38.52(1926/5000)	38.52(1926/5000)
S15850	58.30(2915/5000)	91.48(4574/5000)	91.48(4574/5000)

경성 지연 고장 테스트는 테스트하고자 하는 경로의 의 경로들에 존재하는 지연에 상관없이 테스트를 수행할 수 있는 장점을 가지고 있다. 비록 경성 지연 고장 테스트가 양질의 테스트이기는 하지만, 테스트 가능한 경로 수는 연성에 비해 많이 떨어진다. 만약, 경로의 입력들에 정적 헤저드가 존재하지 않는다는 것이 보장된다면, 강연성 지연 고장 테스트는 경성 지연 고장 테스트로 여겨질 수 있다. 따라서, 표 5에 나타난 강연성 지연 고장 테스트 결과 역시 무시할 수 없다. 만약, 강연성 지연 고장 테스트 역시 생성할 수 없다면, 약연성 지연 고장 테스트를 고려해야 한다. 물론 약연성 지연 고장 테스트는 다른 테스트에 비해 그 질이 떨어지지만, 테스트 가능한 경로 수는 가장 많다. 게다가, 만약 모든 경로의 입력들이 경로 입력 보다 먼저 도착하여 안정화된다는 것이 보장된다면 약연성 지연 고장 테스트로도 지연 고장을 검출할 수 있다. 이러한 테스트 결과는 표 6에 나타나 있다.

표 1에 보인바와 같이, 연성 테스트의 경우, 경성 테스트보다 두 번째 테스트 패턴으로 'X' 값을 많이 갖는다. 따라서 표 3에서 표 6으로 갈수록 지연 고장 검출률은 높아지게 된다.

이러한 결과들은 제안된 알고리즘을 적용할 경우 기존의 스캔 테스트 방법보다 많은 수의 경로들에 대해 고장을 검출할 수 있다는 것을 보여준다. 예를 들어, S15850과 같은 회로에 대해서, 기존의 스캔 기법으로는 1868개의 경로들을 경성으로 테스트할 수 있다. 반면에, 제안된 알고리즘을 적용할 경우, 4049개의 경로들에 대해 경성 지연 고장 테스트를 생성할 수 있다. 강연성의 경우에는, 1882개의 고장들을 가능적 지정 방법으로 테스트할 수 있지만, 새로운 알고리즘으로는 4231개의 고장들을 검출 가능하다. 확장 스캔과 비교할 경우, 제안된 알고리즘을 적용할 경우의 지연 고장 검출률이 거의 확장 스캔과 비슷하게 나타나고 있다. 따라서 새로운 스캔 플립-플롭과 새로운 알고리즘은 높은 지연 고장 검출률과 낮은 면적 오버헤드를 이룰 수 있어 효율적인 방법이라고 할 수 있다.

### VI. 결 론

본 논문에서는 동기 순차 회로의 지연 고장 테스트를 위한 스캔 테스트 기법이 제시되었다. 조합 회로의 내부 입력과 출력은 스캔 사슬을 통해 제어와 관찰 가능하다. 특히, 테스트 용이도는 스캔 플립-플롭에 두 번째 테스트 패턴을 저장함으로써 향상된다. 반면에, 첫 번째 테스트 패턴은 조합 회로 블록의 기능에 의해 생성된다. 첫 번째 테스트 패턴을 효과적으로 생성하기 위해, ATPG를 이용한 방법이 제안되었다. ATPG 도구를 이용하여, 이러한 패턴들을 얻을 수 있을 뿐만 아니라 어떠한 경로들이 역기능적 지정 방법을 통해서만 테스트 가능한 지에 대한 정보도 얻을 수 있다. 이러한 기법들을 실제로 적용하기 위해서, 새로운 스캔 플립-플롭이 제시되었는데, 이 스캔 플립-플롭은 기존의 기능적 지정 방법과 역기능적 지정 방법을 동시에 사용할 수 있어 새로운 테스트 알고리즘을 적용할 수 있다. 각각의 패턴들을 스캔 플립-플롭 안에 저장하기 위해, 두 개의 래치들을 평행하게 놓고 멀티플렉서를 이용하여 엮었다. 따라서 표준 스캔에 비해 추가적인 지연은 멀티플렉서 소자 하나에서만 일어난다.

또한, 스캔 이동 기법을 제안한 테스트 알고리즘의

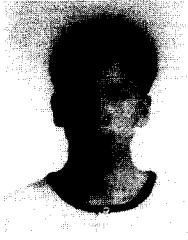


과정에 추가하는 것도 생각해볼 수 있다. 만약, 오직 스캔 이동 방법으로만 검출 가능한 고장 경로가 존재한다면, 추가적인 하드웨어 없이 제안된 스캔 플립-플롭을 이용하여 스캔 이동 기법을 적용할 수 있기 때문이다.

### 참 고 문 헌

- [1] V. Iyengar, B. Rosen, and I. Spillinger, Delay Test Generation Algebra and Algorithms, *Proc. of International Test Conference*, pp. 867-876, 1988.
- [2] J. Waicukauski, E. Lindbloom, B. Rosen and V. Iyengar, Transition Fault Simulation, *IEEE Design and Test*, pp. 32-38, April 1987.
- [3] Hsieh, E. P., Rasmussen, R. A., Vidunas, L. J., Davis, W. T., Delay Test Generation, *Proc. of Design Automation Conference*, pp. 486-491, June 1977.
- [4] Smith, G. L. Model for Delay Faults Based upon Paths, *Proc. of International Test Conference*, pp. 342-349, November 1985.
- [5] A. Saldanha, R. K. Brayton and A. L. Sangiovanni-Vincentelli, Equivalence of Robust Delay-fault and Single Stuck-fault Test Generation, *Proc. of Design Automation Conference*, pp. 173-176, 1992.
- [6] K. T. Cheng and H. C. Chen, Delay Testing for Non-robust Untestable Circuits, *Proc. of International Test Conference*, pp. 954-961, 1993.
- [7] W. Ke and P. R. Menon, Synthesis of Delay-verifiable Combinational Circuits, *IEEE Trans. on Comput.*, vol. 44, pp. 213-222, February 1995.
- [8] Bill Underwood, Wai-On Law, Sungho Kang, Haluk Konuk, Fastpath: A Path-Delay Test Generator for Standard Scan Designs, *Proc. of International Test Conference*, pp. 154-163, 1994.
- [9] M. Schulz et al., Advanced Automatic Test Pattern Generation Techniques for Path Delay Faults, *Proc. of Fault Tolerant Computing Symp.*, 1989.
- [10] J. D. Lesser and J. J. Schedletsy, An Experimental Delay Test Generator for LSI logic, *IEEE Trans. on Comput.*, pp. 235-248, March 1980.
- [11] Sungho Kang, Wai-On Law, and Bill Underwood, Path-Delay Fault Simulation for a Standard Scan Design Methodology, *Proc. of International Conference on Computer Design*, 1994.
- [12] J. Savir, Scan Latch Design for Delay Test, *Proc. of International Test Conference*, pp. 446-453, 1997.

## 저 자 소 개

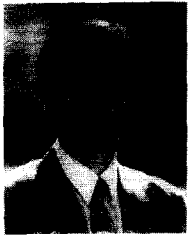


許 堧 會(正會員)

1975년 6월 7일생, 1998년 2월 연세대학교 전기공학과 학사, 1998년~현재 연세대학교 전기공학과 석사 과정, <주관심 분야: Testing and Testable Design, 비동기 회로의 테스트, DSP architecture>

康 容 碩(正會員)

1972년 7월 10일생, 1995년 2월 연세대학교 전기공학과 학사, 1997년 8월 연세대학교 전기공학과 석사, 1997년~현재 연세대학교 전기공학과 박사 과정, <주관심분야: Testing and Testable Design, VLSI & CAD>



姜 成 昊(正會員)

1963년 4월 13일생, 1986년 2월 서울대학교 제어계측공학과 학사, 1988년 5월 The University of Texas at Austin 전기 및 컴퓨터 공학과 석사, 1992년 5월 The University of Texas at Austin 전기 및 컴퓨터공학과 박사, 1989년 11월~1992년 8월 Schlumberger Inc. Research Scientist, 1992년 8월~1994년 6월 Motorola Inc. Senior Staff Engineer, 1994년 9월~현재 연세대학교 기계전자공학부 조교수, <주관심 분야: Testing and Testable Design, VLSI&CAD, Design Verification, VLSI & System Design>