

순차 회로를 위한 효율적인 지연 고장 테스트 알고리즘

허경희, 강용석, 강성호
연세대학교 전기공학과

Efficient Delay Test Algorithm for Sequential Circuits

Kyung-Hoi Huh, Yong-Seok Kang and Sungho Kang
Dept. of Electrical Eng., Yonsei Univ.

Abstract - 지연 고장 테스트는 디지털 회로의 정확한 동작을 보장하기 위해서 필수적이다. 그러나, 순차 회로에는 상태 레지스터들이 존재하기 때문에, 지연 고장을 검출하는 것이 쉽지 않다. 이러한 난점을 해결하기 위해 본 논문에서는 역기능적 지정 방법을 좀 더 효율적으로 적용할 수 있는 테스트 알고리즘을 제안한다. ISCAS89 벤치마크 회로에 대한 실험 결과, 테스트 가능한 경로의 수를 기존의 스캔 기법들에 비해 크게 향상시킬 수 있다는 것을 알 수 있다.

발시켜야 하고 경로상의 모든 신호들은 특정 천이가 도달하기 전까지 최종 값을 알 수 없어야 한다. 경성 테스트의 요건을 갖추지 못한 테스트를 연성 테스트라고 부른다. 경성 테스트에 가까운 연성 테스트를 강연성 (strong non-robust) 테스트라 부른다. 강연성 테스트는 모든 요건이 경성 테스트와 같지만 정적 해저드 (static hazard)가 있어도 무방한 두 패턴의 테스트이다. 그 외에 약연성 (weak non-robust) 테스트(9)도 있지만 테스트 질이 많이 떨어지는 것으로 알려져 있다.

1. 서 론

2. 본 론

디지털 회로가 빨라지고 집적되면서, 회로를 좀 더 효과적으로 테스트하는 것이 중요하게 되었다. 게다가, 회로의 기능뿐만 아니라, 신호 천이 지연을 검사하는 것이 중요시되면서, 제조 공정에서 생긴 결함을 검출하기 위한 고착 고장과 지연 고장의 테스트가 많이 사용되어 왔다. 이러한 지연 고장 테스트의 목적[1]은 생산된 회로가 요구되는 클럭 주파수 내에서 제대로 동작하는 지를 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)[2, 3]과 경로 지연 고장 모델 (path delay fault model)[4, 5]이다. 게이트 지연 고장 모델은 지연이 회로 내의 게이트 입력이나 출력에 집중되었다고 가정하는 것이므로 회로내의 각 경로에 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 기초를 둔 테스트는 회로의 경로에 따른 집중 또는 분산된 결함을 모두 검출할 수 있다. 그러나, 경로의 수는 회로의 크기에 따라 지수 함수적으로 증가하기 때문에, 회로내의 모든 경로에 대한 테스트는 사실상 불가능하다.

경로 지연 고장을 위한 테스트는 경로에 천이를 발생시키고, 발생한 천이를 경로를 따라 전파시킬 수 있는 벡터 쌍으로 이루어진다. 여기서, 발생한 천이가 경로의 끝까지 전파되는 동안 걸리는 시간이 그 경로의 지연 시간이다. 조합 회로를 위한 경로 지연 고장 테스트 방법과 테스트를 위한 설계 방법에 대해서는 그 동안 꾸준한 연구가 진행되었다[5, 6, 7].

경로 지연 고장에 대한 테스트에는 크게 무해저드 경성 (hazard free robust) 테스트, 경성 (robust) 테스트, 연성 (non-robust) 테스트의 세 종류로 나눌 수 있다[8]. 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 유

2.1 스캔 테스트 방법

일반적으로, 표준 스캔 (standard scan)을 사용하는 테스트 방법으로는 스캔 이동 (scan shifting) 방법[1]과 기능적 지정 (functional justification) 방법이 있다 [9]. 스캔 이동 방법의 경우에는 테스트 가능한 많은 경로에 대한 테스트를 생성하지 못할 수가 있다. 이러한 스캔 이동의 문제점을 해결하고 정확한 지연 고장 검사를 위해 기능적 지정 방법이 사용된다. 이 방법에서 첫 벡터는 스캔 플립-플롭으로 입력해 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. 또한, 조절 용이도가 높은 첫 번째 시간대에 필요한 테스트 패턴들의 논리 값들은 표 1에서 보는 바와 같이 '0'이나 '1'로 확정되어 있지만은 않아 스캔 입력을 낭비하기도 한다.

표 1. 경로 지연 고장 테스트에서의 경로와 입력

소자 유형	경로 입력 천이	경로의 입력 조건	
		경성	연성
AND	상승 천이	X1	X1
/NAND	하강 천이	11(무해저드)	X1
OR	상승 천이	00(무해저드)	X0
/NOR	하강 천이	X0	X0

확장 스캔 플립-플롭 (enhanced scan flip-flop)[7]을 이용하면 모든 벡터 쌍을 조절할 수 있어 이러한 문제점을 해결할 수 있다. 즉, 지연 고장 테스트 용이도는 크게 증가한다. 그러나 확장 스캔 플립-플롭은 추가되는 플립-플롭만큼의 면적이 더해지므로, 면적 오버헤드가 너무나 커서 실제 순차 회로에는 거의 사용되지 못하고 있다.

이러한 스캔 테스트 기법들의 단점을 보완하기 위해 새로운 스캔 구조와 역기능적 지정 (reverse functional justification) 방법이 [10]에 제시되었다. 역기능적 지정 방법은 기능적 지정 방법에서 조절 용이도가 높은 스캔 플립-플롭을 'X'로 낭비하는 것을 막고자 두 번째 테스트 패턴을 스캔 플립-플롭에 저장하는 방식이다. 반면에, 첫 번째 테스트 패턴은 회로의 기능에 의해 지정되는 방법이므로, 첫 번째 테스트 패턴을 효율적으로 생성

하는 알고리즘이 필요하다. 그러나, 이를 위해 [10]에서 제시된 방법은 첫 번째 테스트 패턴 생성률을 크게 떨어뜨려 결국 지연 고장 검출률을 원하는 수준으로 높이지 못했다. 이를 보완하기 위해, 본 논문에서는 주입력과 스캔 플립-플롭을 모두 조절하여 첫 번째 테스트 패턴을 생성하는 새로운 알고리즘을 개발하였다.

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 입력 패턴을 $P0$ ($Pp0$ 과 $Ps0$ 으로 구성)으로 표시한다. 출력의 경우에는, $R1$ 과 $R2$ 를 $P1$ 과 $P2$ 의 응답 패턴을 의미하는 것으로 각각 나타낸다.

[10]에서와 같이 $Ps1$ 에 따라 회로를 수정할 수 있고, 수정된 회로에 'X' 제약 조건 없이 ATPG를 수행하면 $P0$ 을 얻을 수 있다. $Ps1$ 을 생성하기 위해 [10]에서는 오직 $Pp0$ 만을 조절하였지만, 본 논문은 $Pp0$ 뿐만 아니라 $Ps0$ 까지 조절하는 방법을 제시한다.

2.2 테스트 알고리즘

$Ps1$ 을 얻기 위한 새로운 방법은 $Pp0$ 과 $Ps0$ 을 동시에 조절하는 것이기 때문에, 첫 번째 시간 프레임의 플립-플롭에 $Ps2$ 뿐만 아니라 $Ps0$ 도 스캔 사슬(scan chain)을 이용하여 넣어준다. 즉, $Ps2$ 는 L3-스캔 이동 모드를 통해 그림 1[10] 스캔 구조의 L3으로 저장되고, $Ps0$ 은 L2-스캔 이동 모드를 통해 L2로 입력 들어간다. $Ps1$ 을 생성하기 위해, $Pp0$ 이 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 하게 된다. 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 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 천이가 일어났는지를 확인할 수 있다. 클럭과 $test_opt$ 신호에 따른 전체적인 테스트 순서는 그림 2에 나타나 있다.

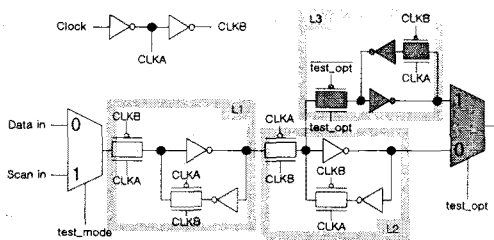


그림 1. 제시된 스캔 플립-플롭

표 2. 동작 모드의 종류

test mode	test opt	동작 모드
0	0	정상 동작 모드
0	1	클럭 모드
1	0	L2-스캔 이동 모드
1	1	L3-스캔 이동 모드

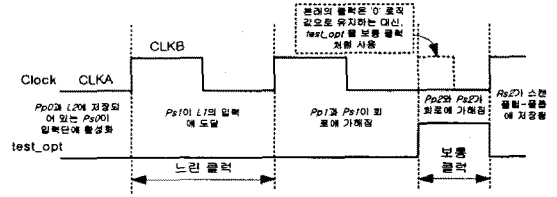


그림 2. 테스트 순서

그림 1의 스캔 구조는 $test_opt$ 신호를 '0'으로 유지함으로써 기존의 기능적 지정 방법을 적용할 수 있다. 따라서, 지연 고장을 효과적으로 검출하기 위해 기능적 지정 방법과 역기능적 지정 방법을 같이 사용하는 새로운 테스트 알고리즘을 제안한다. 이러한 알고리즘은 그림 3에 제시되어 있다. 우선, 역기능적 지정 방법을 시행하기 위해 순차 회로의 조합 회로 블록에 대해 모든 가능한 지연 고장 테스트를 생성한다. 얻어진 첫 번째 테스트 패턴 중에 회로의 기능에 의해 생성 가능한 경로가 있는지를 ATPG를 이용한 방법으로 조사하고, 있다면 가능한 경로를 기록한다. 이 과정을 통해 $P0$ 을 얻을 수 있으므로, $P0$, $P1$, $P2$ 를 순차적으로 회로에 가해 회로의 경로 지연 고장을 테스트할 수 있다.

```

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

/* ----- Functional justification sequence ----- */
{
  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 each path (Path(i))
      to the sequential circuit.
    }
  }
}

```

그림 3. 테스트 알고리즘

기능적 지정 과정은 선택적이다. 즉, 만약 위의 역기능적 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 역기능적 지정 과정에서 테스트 불가능한 경로가 있다면 해당하는 테스트 패턴을 회로에 가해 테스트를 수행할 수 있다. 이것은 역기능적 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 정확히 회로를 테스트 하고자 한다면, 전체적인 테스트 알고리즘을 통해 회로의 정확한 동작을 보장할 수 있을 것이다.

2.3 실험 결과

표 3과 표 4는 ISCAS89 순차 벤치마크 회로에 대해 각각 경성과 강연성으로 경로 지연 고장 실험을 한 결과이다. 괄호 안의 두 수는 전체 경로에 대해 테스트 가능한 경로 수를 의미한다. 경로 수가 5000개 이하인 회로들에 대해서는 모든 경로들에 대해 고려하였고, 5000개를 넘는 큰 회로들에 대해서는 5000개의 임의 경로에 대해 고려하였다.

표 3. 경성 테스트 결과

벤치마크 회로	기능적 지정	확장 스캔	새로운 테스트 알고리즘
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 4. 강연성 테스트 결과

벤치마크 회로	기능적 지정	확장 스캔	새로운 테스트 알고리즘
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)

이 결과들을 통해 본 논문에서 제안한 새로운 테스트 알고리즘이 기존의 스캔 테스트 기법들보다 많은 수의 경로 지연 고장을 검출할 수 있다는 것을 알 수 있다. 예를 들어, S15850 회로가 기존의 스캔 플립-플롭을 사용하여 설계되었다면, 위의 실험에서 선택된 5000개의 경로 중에 오직 1868개의 경로만이 기능적 지정 방법으로 경성 테스트가 가능할 것이다. 그러나, 반대로 만약 새로 고안된 스캔 플립-플롭을 가지고 설계되었다면, 새로운 테스트 알고리즘을 적용하여 4049개의 경로가 테스트 가능할 것이다.

또한, 새로운 테스트 알고리즘과 확장 스캔 기법과 비교해 볼 때, 고장 검출률 측면에서는 거의 같은 수준을 유지하는 것을 알 수 있다. 반면에 면적 오버헤드는 표준 스캔 플립-플롭과 비교해 볼 때 46% 정도로 확장 스캔의 80%에 비해 아주 적다. 따라서 새로운 스캔 플립-플롭과 테스트 알고리즘은 적은 면적 오버헤드로 높은 고장 검출률을 갖는 효율적인 경로 지연 고장 테스트 방법으로 사용될 수 있을 것으로 기대된다.

3. 결 론

본 논문에서 제안한 스캔 테스트 기법은 동기 순차 회로의 효율적인 지연 고장 검출을 위해 개발되었다. 테스트 용이도는 두 번째 테스트 패턴을 스캔 플립-플롭에 저장하는 역기능적 지정 방식을 통해 향상되었다.

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] S. Reddy, C. Lin and S. Patil, "An Automatic Test Pattern Generator for the Detection of Path Delay Faults," *Proc. of International Conference Computer-Aided Design*, pp. 284-287, 1987.
- [6] I. Hamzaoglu, J. Patel, "Compact Two-Pattern Test Set Generation for Combinational and Full Scan Circuits," *Proc. of International Test Conference*, pp. 944-953, 1998.
- [7] S. Dasgupta, R. G. Walther and T. W. Williams, "An enhancement to LSSD and some applications of LSSD in reliability, availability and serviceability," *Proc. of Fault Tolerant Computing Symp.*, pp. 32-34, 1981.
- [8] B. Underwood, W. Law, S. Kang, H. Konuk, "Fastpath: A Path-Delay Test Generator for Standard Scan Designs," *Proc. of International Test Conference*, pp. 154-163, 1994.
- [9] M. Schulz, F. Fink and K. Fuchs, "Parallel Pattern Fault Simulation of Path Delay Faults," *Proc. of Design Automation Conference*, pp. 357-363, 1989.
- [10] 허경희, 강용석, 강성호 "순차 회로의 지연 고장 검출을 위한 새로운 스캔 설계", 전기학회논문지, 48권 9호 A, pp.1161-1165, 1999.