

고장 선택 테이블을 사용한 다중 고착 고장 진단 방법

A Multiple Fault Diagnosis Method Using a Fault Selection Table

임요섭, 이주환, 강성호
연세대학교 전기전자공학과
yoseop@soc.yonsei.ac.kr, {eldsich, shkang}@yonsei.ac.kr

Abstract

With the increasing complexity of VLSI devices, more complex faults have appeared. Most of previous fault diagnosis methods considered a single defect assumption. However, for present technologies and chip sizes, defects have tendency to be clustered. So, we propose a multiple fault diagnosis method using a fault selection table. The proposed method can diagnose multiple defects based on a single fault simulator. In spite of a multiple fault diagnosis, the number of candidate faults is drastically reduced. Experimental results for ISCAS85 benchmark circuits prove the efficiency of the proposed method.

I. 서론

VLSI(Very Large Scale Integration)의 복잡도가 증가하면서, 고장 진단의 요구 또한 증대되었다. 고장 진단이란 동작의 오류를 일으키는 결함의 위치와 종류를 추론해내는 과정을 말한다. 정확한 고장 진단으로 설계와 공정의 오류를 수정할 수 있고, 이는 수율 향상으로 이어진다. 따라서 칩의 품질을 높이고 생산 비용을 절감하기 위한 효율적인 고장 진단 방법론을 개발하는 것은 매우 중요한 일이다.

많은 고장 진단 방법론들은 단일 결함을 가정하였다. 그러나 현재의 공정 기술과 칩의 크기를 고려하였을 때에는 이런 가정이 사실이라고 보기 어렵다. 오히려, 다중 결함 가정이 보다 정확히 사실을 반영한다고 말할 수 있다.

최근에 다중 고장 진단 방법론에 대한 논문들이 발표되었다[1-3]. [1]에서는 다중 고장이 존재하는 회로에서 한 개의 고장만 활성화하여 전파하는 SLAT 패턴들의 성질을 활용하여 전체의 고장이 관찰된 패턴들을 설명할 수 있는 최소의 고장 집합을 구하는 방법을 제시하였다. 그러나 실제의 회로에는 SLAT 테스트로 검출할 수 없는 고장들이 존재하며, 진단된 고장 집합의 크기가 큰 단점을 가지고 있다.

[2]에서는 경로 추적 기법을 사용하여 초기의 고장 후보군을 결정한 후, 랜덤 패턴을 인가하여 고장 시뮬레이션을 수행하여 실제 결함 회로의 출력 결과와 유사도가 높은 후보 고장을 선택한다. 이렇게 선택한 고장을 회로에 삽입하고 다중 고장 시뮬레이션을 반복적으로 수행하여 고장 후보 집합을 결정한다.

[3]의 논문에서는 [2]의 반복적으로 다중 고장 시뮬레이션을 수행하고 비교하는 방법이 시간이 많이 소요되는 문제를 보완하기 위하여 여러 기법과 다중 고장 시뮬레이터를 적용하여 고장 진단을 수행하나 이 역시 다중 고장 시뮬레이터 사용하므로 수행 시간이 회로 크기의 증가에 따라 기하급수적으로 증가하는 문제가 존재한다. 이런 문제로 인하여 회로의 크기가 큰 상용 회로에 적용하기는 어렵다.

이 논문에서 제안하는 다중 고착 고장 진단 알고리즘은 단일 고장 시뮬레이션 환경에서 고장 선택 테이블을 사용하여, 적은 수의 고장 집합으로 정확하게 다중 고착 고장 진단을 수행할 수 있다.

2장에서는 제안하는 다중 고착 고장 진단 알고리즘에 대해서 설명하고 3장에서 실제 실험결과를 제시하며, 4장에서 논문의 결론을 정리하였다.

II. 제안하는 알고리즘

제안하는 다중 고착 고장 진단 알고리즘은 단일 고착 고장 시뮬레이터를 사용하기에 회로 크기의 증가에 따라 수행 시간이 기하급수적으로 증가하는 문제점이 발생하지 않는다.

실제 결함 회로의 출력 결과의 오류를 경로 추적 기법으로 분석하여 초기 고장 후보군을 결정한다. 초기 고장 후보군을 단일 고착 시뮬레이션을 수행하며 동시에 고장 선택 테이블을 생성한다. 시뮬레이션 종료 후에 고장 후보 집합 결정 알고리즘을 수행하여 최종 고장 진단 결과를 출력한다.

고장 선택 테이블에는 고장 후보 선택 기준이 되는 두 가지 정보가 저장된다. 하나는 각 패턴별로 후보 고장의 시뮬레이션 결과와 실제 결함 회로의 출력 결과의 유사도를 정량적으로 나타내는 고장 점수와 다른 하나는 후보 고장의 시뮬레이션 결과와 실제 결함 회

로의 출력 결과의 오류의 형태가 동일한 경우를 나타내는 VI이다. VI는 Vector-wise Intersection을 줄여서 나타낸 것으로 [4]에서 정의하였고, [4]의 고장 진단 알고리즘에서 가장 가중치가 큰 기준이며 [1]에서도 explain fails table을 생성하는 데 중요한 요소가 된다.

고장 점수는 [5]에서 제안하였던 방법으로 [4]에서 정의하였던 Vector-wise Intersection, Intersection, Non-prediction과 Mis-prediction의 개념을 확장하여 후보 고장의 시뮬레이션 결과와 실제 결함 회로의 출력 결과의 유사도를 정량적으로 나타낼 수 있는 방법이다. 후보 고장의 시뮬레이션 결과나 실제 결함 회로의 출력 결과의 한 출력단에서만 오류가 관찰된다면 1점을 감소시키며, 동시에 두 곳에 모두 오류가 관찰된다면 1점을 증가시킨다. 또한, VI가 나타날 경우에는 주출력단의 수만큼 점수를 증가시킨다.

시뮬레이션 종료 후에 생성된 고장 선택 테이블을 사용하여 다음과 같은 과정으로 최종 고장 후보 집합을 결정하게 된다.

- 1) 고장 시뮬레이션을 수행한 초기 고장 후보군들과 테스트 패턴들에 대하여, 고장 점수를 계산하고 VI가 나타나는지를 체크한다. 또한, 각 고장별로 테스트 패턴들에 대한 고장 점수를 모두 합한 전체 고장 점수도 계산한다.
- 2) VI가 가장 많이 나타나는 고장들 중에서 전체 고장 점수가 가장 높은 고장을 선택하여 최종 고장 후보 집합에 포함시킨다. 만일 동순위가 있다면 모두 최종 고장 후보 집합에 포함시킨다.
- 3) 최종 후보 고장 집합에 존재하는 고장에서 VI가 나타나는 경우의 테스트 패턴에 대하여 "후보 고장이 실제 결함 회로가 오류를 출력하는 테스트 패턴을 설명한다." 라고 하며 전체 고장 점수를 계산하고 VI의 수를 셀 경우에 이들 패턴을 포함시키지 않도록 체크한다.
- 4) 최종 후보 고장 집합에 포함된 고장들이 전체 고장 점수를 계산할 경우에 이들 고장이 포함되지 않도록 체크한다.
- 5) 3)과 4)단계에서 체크된 고장과 테스트 패턴을 제외하여 전체 고장 점수를 계산한다.
- 6) 2)~5)단계를 반복하며, 만일 2)단계에서 최대 나타나는 VI의 수가 0이라면 최종 고장 후보 집합 결정 알고리즘을 종료한다. 또한, 3)단계에서 실제 결함 회로가 오류를 출력하는 모든 테스트 패턴이 설명되었다면 고장 후보 집합 결정 알고리즘을 종료한다.

위의 알고리즘에서 최종 고장 후보 집합에 포함된 고장들이 설명하는 오류 테스트 패턴의 경우를 제외하

여 전체 고장 점수를 계산하는 이유는, 이미 최종 고장 후보 집합에 포함된 고장들로 인한 영향이 다음 전체 고장 점수 계산 시에 적게 반영되게 하기 위함이다.

III. 실험 결과

실험에 사용한 회로는 ISCAS85회로의 Verilog 버전을 삼성 STD150 라이브러리를 합성하였다. 또한, 테스트 패턴은 Synopsys사의 TetraMAX를 사용하여 생성하였다. 실험은 고착 고장 2개가 삽입한 회로를 30번씩 10개의 회로에 대하여 진행하였다.

표 1. ISCAS85 회로의 고장 진단 결과

회로	SLAT [1]		제안하는 알고리즘	
	진단된 고장의 수	후보 고장의 수	진단된 고장의 수	후보 고장의 수
c432	1.67	9.77	1.67	4.23
c499	1.23	31.27	1.37	3.30
c880	1.90	9.57	1.93	2.70
c1355	1.13	19.93	1.27	3.70
c1908	1.20	10.07	1.30	2.33
c2670	1.83	21.97	1.87	2.67
c3540	1.77	9.53	1.80	3.20
c5315	1.83	14.03	1.87	3.00
c6288	1.70	27.23	1.77	3.62
c7552	1.83	37.83	1.87	3.30
평균	1.61	19.12	1.67	3.20

진단된 고장의 수는 삽입한 2개의 고장 중에서 후보 고장 집합에 포함된 고장들의 수의 평균을 나타낸 것이며, 후보 고장의 수는 후보 고장 집합에 포함된 평균 고장의 수를 나타낸 것이다. SLAT의 결과는 여러 tuple에 존재하는 고장들을 중복하지 않고 세어 나타내었다. 표 1에서, 제안하는 알고리즘은 SLAT보다 진단한 후보 고장의 수가 1/6정도이나 보다 정확하게 다중 고착 고장을 진단하는 것을 확인할 수 있다.

IV. 결론

이 논문에서 고장 선택 테이블을 사용한 다중 고착 고장 진단 방법을 제안하였다. 제안한 알고리즘은 단일 고장 시뮬레이션 환경에 적용되어 수행 시간이 회로 크기의 증가에 선형적으로 비례하여 증가되므로 다른 다중 고장 진단 방법론들이 가진 기하급수적인 증가 문제를 해결하였다. 또한, 적은 수의 후보 고장으로 정확하게 고장을 진단하는 것을 3장의 실험 결과로 확인할 수 있다.

Acknowledgment

본 논문은 IDEC(IC Design Education Center)의 CAD tool 지원을 받은 것임.

참고문헌

- [1] L. M. Huisman, "Diagnosing Arbitrary Defects in Logic Designs Using Single Location at a Time (SLAT)," *IEEE trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 23, no. 1, pp. 91-101, 2004.
- [2] J. B. Liu and A. Veneris, "Incremental Fault Diagnosis," *IEEE trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 2, pp. 240-251, 2005.
- [3] Z. Wang, M. Marek-Sadowska, K. Tsai and J. Rajski, "Analysis and Methodology for Multiple-Fault Diagnosis," *IEEE trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 25, no. 3, pp. 558-575, 2006.
- [4] S. Venkataraman and S. Drummonds, "Poirot : Applications of a Logic Fault Diagnosis Tool," *IEEE Design & Test of Computers*, pp. 19-30, 2001.
- [5] Yoseop Lim, Joohwan Lee, Hyungjun Cho and Sungho Kang, "An Efficient Diagnosis Algorithm for Multiple Stuck-at Faults," *Proc. of SOC Design Conference*, pp. 394-397, 2006.