

빠른 고장 진단을 위해 고장 점수를 이용한 고장 탈락 방법

A fault dropping technique using fault scores for fast fault diagnosis

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

Abstract

In order to support time-to-market, the demand for fast fault diagnosis has been increased. We propose a fault dropping technique using fault scores for fast fault diagnosis. Experimental results using the ISCAS benchmark circuits show that the efficiency of the fault dropping technique.

I. 서론

VLSI 회로와 디지털 시스템의 복잡도가 나날이 증가하고 크기가 줄어들어 따라, 고집적 회로의 고장 분석(fault analysis)이 점점 더 어려워지고 있다. 따라서 고장 진단(fault diagnosis)의 중요성이 꾸준히 늘어나고 있으며, 빠르게 변화하는 시장의 요구를 만족시키기 위해서는 짧은 시간 안에 고장 진단이 이루어져야 한다.

대부분의 고장 진단은 고장 시뮬레이션(fault simulation)을 이용하며, 고장 진단은 고장 시뮬레이션 과정에서 많은 시간을 소비한다. 그러므로 고장 진단을 수행하는 시간을 절약하기 위해서는 고장 시뮬레이션의 실행 시간을 줄이는 것이 큰 도움이 된다. 같은 회로를 대상으로 했을 경우에, 고장 시뮬레이션의 실행 시간은 테스트 패턴(test pattern)의 수와 고장 후보(candidate fault)의 수에 의하여 결정된다.

정확한 고장 진단을 수행하기 위해서는 각각의 고장 후보 간의 동일한 횟수의 비교를 통해 [5]와 같은 매칭 알고리즘(matching algorithm)을 이용하여 실제 고장 후보의 순위를 결정해야 하며, 또한 동 순위 후보 고장의 개수를 줄이기 위해서 모든 테스트 패턴의 정보를 이용하는 것이 필요하다. 따라서 테스트 패턴의 수를 조절하여 고장 시뮬레이션 시간을 줄이는 것은 고장 진단 결과에 좋지 않은 영향을 미친다.

고장 후보의 수를 줄이기 위해서 [1], [2], [3], [4]의 방법이 제안되었다. [1]에서는 고장 시뮬레이션을 시작하기 전에 고장 후보의 수를 미리 줄여 주었으며, [2], [3], [4]에서는 고장 시뮬레이션을 수행하는 과정 중에 후보 고장의 수를 줄여 주었다. 하지만, [2], [3],

[4]와 같은 방법은 테스트 대상 회로 내의 고장 유무를 판별할 수는 있지만, 고장의 위치와 종류를 밝혀내는 고장 진단을 수행할 수는 없다.

본 논문을 통하여, 매칭 알고리즘을 사용하여 고장 시뮬레이션 중에 고장을 탈락(fault dropping)시켜 고장 진단의 수행 시간을 줄이는 방법을 제안하고자 한다.

이 논문은 다음과 같이 구성되어 있다. II 절에서 고장 진단 시에 고장을 탈락시키는 방법에 대해 제안할 것이며, III 절에 실험 결과를 제시할 것이다. 그리고 IV 장에서 결론을 짓도록 하겠다.

II. 본론

테스트의 경우에는 고장의 유무만을 판단하면 되므로, 테스트 패턴을 적용하였을 때에 해당 고장 후보가 검출(detected)되면 고장 탈락이 일어난다. 하지만, 고장 진단은 고장의 유무만을 판별하는 것이 아니라, 고장의 위치와 종류를 밝혀내야 하므로 테스트 패턴이 고장 후보를 한 번 검출했다고 해서 고장을 탈락시킬 수 없다.

고장 시뮬레이션을 통한 고장 진단은 최종적으로 매칭 알고리즘을 통한 점수 매기기와 매겨진 점수를 바탕으로 한 순위 결정을 통해 이루어진다. 이 때의 고장 진단 결과를 살펴보면, 실제 고장이 있는 위치와 가까운 고장 후보일수록 점수가 높으며, 그렇지 않을수록 점수가 낮게 나타난다. 또한, 실제 고장의 위치와 멀리 떨어져 있어서 서로 영향을 미칠 확률이 아주 적은 고장 후보는 상당히 낮은 점수를 가지게 된다.

다른 고장 후보와 비교해서 매우 낮은 점수를 갖기 위해서는 지속적으로 음의 값이 더해져야 한다. 그러므로 실제 고장이 아닐 확률이 높은 고장 후보는 모든 테스트 패턴에 대해서 고장 시뮬레이션을 실시하지 않아도 결과를 예상하는 것이 가능하다.

따라서 고장 시뮬레이션을 모든 테스트 패턴에 대해서 실시한 이후에 매칭 알고리즘을 적용하여 각각의 고장 후보에 해당하는 점수를 구하는 것이 아니라, 하나의 테스트 패턴을 고장 후보에 가할 때마다 고장 점수를 구하여 실제 고장일 확률이 현저히 낮은 고장을 탈락시키면 보다 빠른 고장 진단을 수행할 수 있다.

III. 실험 결과

모든 실험은 ISCAS85 회로 4개와 ISCAS89 회로 6개를 이용하여 SUN Fire 880 서버에서 이루어 졌다. ISCAS89 회로들은 모두 완전 주사 회로이며, 테스트 패턴은 TetraMAX를 이용하여 구했다. 대상 회로에 1개의 고장과 2개의 고장을 각각 30회씩 임의로 삽입하여 실험을 진행하였다.

고장 탈락을 적용한 고장 진단 결과와 고장 탈락을 적용하지 않은 고장 진단 결과를 표 1에서 비교하였다.

표 1. 고장 탈락을 적용한 고장 진단 결과와 고장 탈락을 적용하지 않은 고장 진단 결과의 비교

회로	1개의 고장 삽입				2개의 고장 삽입			
	미적용		고장탈락 적용		미적용		고장탈락 적용	
	FHR	CPU time	FHR	CPU time	FHR	CPU time	FHR	CPU time
C3540	1.00	2.57	1.00	1.22	1.03	4.40	1.03	1.74
C5315	1.00	1.90	1.00	1.55	1.00	3.64	1.13	2.75
C6288	1.00	14.14	1.00	5.99	2.20	16.5	2.83	6.77
C7552	1.00	4.11	1.00	2.67	1.63	5.94	1.43	4.15
S5378	1.00	11.77	1.00	6.97	1.00	16.47	1.50	9.16
S9234	1.00	19.70	1.00	11.85	1.00	41.66	1.03	15.25
S13207	1.00	42.12	1.00	21.53	1.17	59.70	1.67	22.87
S15850	1.00	30.96	1.00	18.65	1.30	93.72	1.33	33.05
S35932	1.00	16.07	1.00	13.09	1.00	20.94	1.17	16.62
S38584	1.00	180.7	1.00	114.6	1.00	387.7	1.14	132.8

표 1에서 FHR(first hit rank)은 실제로 삽입한 고장을 고장 진단의 결과에서 처음으로 찾아내려면 몇 개의 고장을 시험해야 하는 지를 나타낸다. 또한, CPU time은 전체 고장 진단의 수행 시간을 나타낸다.

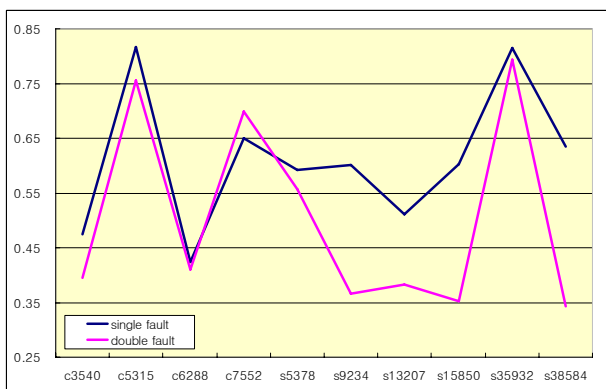


그림 1. 고장 탈락을 적용한 고장 진단 시간과 고장 탈락을 적용하지 않은 고장 진단 시간의 비율

표 1에서 모든 회로의 CPU time이 고장 탈락을 적

용하지 않았을 때보다 고장 탈락을 적용하였을 경우에 짧은 것을 확인할 수 있다. 보다 정확한 CPU time의 비교를 위해서 그림 1에 고장 탈락 유무에 따른 비율을 나타내었다. 고장 탈락을 적용하지 않았을 때보다 고장 탈락을 적용하여 고장 진단을 수행하였을 경우의 CPU time이 최대 66% 줄어 드는 것을 확인할 수 있다.

IV. 결론

빠르게 변화하는 시장의 요구에 부응하는 고집적 회로를 제작하기 위해서 짧은 수행 시간을 가지는 고장 진단이 필요하다. 빠른 고장 진단을 위하여 고장 점수를 이용하여 고장 시뮬레이션 과정 중에 고장을 평가하여 실제 고장일 것 같은 확률이 낮은 고장을 탈락시키는 방법을 제안하였다. 고장 탈락을 적용한 고장 진단이 ISCAS 회로들을 이용하여 검증되었으며, 고장 탈락을 적용하여 고장 진단을 수행하였을 경우에 적용하지 않은 경우의 56%의 고장 진단 실행 시간을 평균적으로 가진다. 또한, 고장 시뮬레이션을 수행할 때의 테스트 패턴 순서를 변화시키는 것을 통해 더 나은 결과를 얻을 수 있을 것으로 기대된다.

참고문헌

- [1] P.R. Menon, Y. Leventel and M. Abramovici, "Critical path tracing in sequential circuits", in *Proc. of IEEE International Conference on Computer-Aided Design*, pp. 162–165, 1988.
- [2] A. Verreault, E.M. Aboulhamid and Y. Karkouri, "Multiple fault analysis using a fault dropping technique", in *Proc. of International Fault-Tolerant Computing Symposium*, pp. 162–169, 1991.
- [3] S. Gai and P.L. Montessoro, "The fault dropping problem in concurrent event-driven simulation", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 968–971, 1991.
- [4] Y. Karkouri, E.M. Aboulhamid, E. Cerny and A. Verreault, "Use of fault dropping for multiple fault analysis", *IEEE Transactions on Computers*, pp. 98–103, 1994.
- [5] Joohwan Lee, Yoseop Lim, Hyungjun Cho and Sungho Kang, "An Accurate Matching Algorithm with essential factors for Fault Diagnosis", in *Proc. of International SOC design Conference*, pp. 301–304, 2006.