

경계스캔이 적용된 보드에서의 고장 모델 및 진단 기법

문 권 우* , 송 오 영*
*중앙대학교 전자전기공학부
e-mail:song@jupiter.cie.cau.ac.kr

Fault Models and Diagonousis of Boundary Scan Board

Kweon Woo Moon*, Ohyoung Song*
*School of Electrical Electronic Engineering, Chung-Ang University

요약

최근에 생산되는 디지털 VLSI칩들은 그 집적도가 계속 높아지고 있으며, 이러한 칩들을 장착한 보드의 경우도 그 복잡성이 점차 높아지고 있다. 이에 따라 칩 및 보드에 대한 철저한 테스트 과정이 요구된다. 지금까지 보드 테스트 방법으로 널리 쓰였던 ICT(In-Circuit Test)는 칩의 고집적화에 따른 핀간격의 조밀화와 SMT(Surface Mount Technology), BGA(Ball Grid Array), MCM(Multi Chip Module) 등의 새로운 패키징 방식의 등장에 따라 테스트 방법으로서의 한계성을 드러내고 있다. 이에 대한 대안으로 등장한 IEEE Std 1149.1 은 ICT의 한계성을 극복할 수 있는 기술일 뿐 아니라 여러 가지 장점을 가지고 있으며 그 활용 분야도 다양하다. 본 논문에서는 IEEE Std 1149.1에 따라 설계된 보드 상에서 발생 가능한 고장들에 대한 고장 모델을 제시한다. 또한 각 고장 모델들의 양상과 진단 기법을 제시한다. 이를 통해 IEEE Std 1149.1에 따라 설계된 보드 상에서 발생한 고장들을 검출할 수 있으며, 고장의 종류 및 성격, 그리고 고장의 발생 위치 등의 정보를 얻을 수 있다. IEEE Std 1149.1에 따른 보드 설계가 보드의 신뢰성 보장에 긴요함을 인식하는 계기가 되기를 기대하며 제시된 고장 모델 및 진단 기법이 기술적으로 중요한 참고자료가 되기를 기대한다.

I. 서 론

최근에 생산되는 VLSI 칩들은 그 성능이 우수할 뿐 아니라 저전력소모등을 위한 이유로 점차로 소형화되고 있다.

하지만 새로운 패키징 방식이 적용된 칩들이 등장함에 따라 탐침을 접촉시킬 수 없는 경우가 많아졌다. 이를 해결하기 위하여 세계의 유명 시스템 회사들은 자사에서 생산한 보드의 테스트를 위해 ICT의 한계를 극복할 수 있는 테스트 기술을 개발해 사용해 왔다. 1985년 세계 주요 보드 생산 업체들이 JTAG이라는 그룹을 결성하고, 보드 테스트의 어려움들을 해결할 수 있는 기술을 제시하였다. 이 기술이 1990년 IEEE Std 1149.1로 표준화되어“경계 스캔 테스트”라 불리게 되었다. 경계 스캔이 적용된 칩은 추가적인 4개의 핀을 가지며, 칩 내부에 핀과 코어 로직 사이에 추가적인 경계스캔 셀이 있으며, 경계 스캔 테스트 수행을 위한 몇 개의 추가 로직이 존재한다.

경계 스캔 구조는 보드 테스트 뿐만 아니라 다양한 분야에 적용되고 있다. 그 대표적인 예로는 ISP(In Sytem Programming), Hardware Debugger 등을

들 수 있겠다. 또한 기존의 ICT 테스트가 생산 시 보드의 고장 유무를 테스트하는 용도에 국한되어있던 반면, 경계 스캔 테스트는 사용 중에, 현장에서 쉽고 빠른 시간 내에 보드의 이상 유무를 검사할 수 있어 유지 보수에 있어서도 편리함을 제공한다.

본 논문에서는 15가지의 고장 모델을 제시한다. 보드 상에서 발생하는 고장은 크게 3가지 - 고착 고장, 단락 고장, 연결 고장 - 으로 분류되며, 한 연결선에 이들 고장이 오로지 하나만 발생하는 단순 고장 모델로부터 한 연결선에 이들 고장이 동시에 여러 개 발생하는 복합 고장 모델들에 대해 다루고 있다. 보드 상에서 발생할 수 있는 고장 양상들은 무수히 많다. 본 논문에서 제시하는 고장 모델들은 실용적인 면에서 최적의 고장 모델들이 될 것으로 생각된다.

II. 경계스캔테스트에 사용된 패턴들

2.1 고착 고장 검출용 패턴

고착 고장이란 보드상의 특정 넷이 VCC 또는 GND와 연결되어 어떤 값을 넷에 인가하더라도 고정값이 읽혀지는 고장을 말한다. 1-고착 고장이란 특정 넷이 VCC와 연결되어 항상 넷에서 1값이 읽혀지는 고장을 말하며, 0-고착 고장이란 특정 넷이

GND와 연결되어 항상 넷에서 0값이 읽혀지는 고장을 일컫는다.

이러한 고착 고장을 검출하기 위해서는 모든 출력 경계스캔 셀에 '0'을 인가하고, 그 후 다시 모든 출력 경계스캔 셀에 '1'을 인가하면 된다. 0-고착 고장이 발생했다면 해당 넷에 연결된 경계스캔 센스 셀에는 두 경우 모두에 대하여 '0'값이 저장될 것이고, 1-고착 고장이 발생했을 경우에는 두 경우모두에 대하여 '1'값이 저장되어 고장 유무를 쉽게 검출해 낼 수 있다.

2.2 LOG(N) 연결 고장 검출용 패턴

연결 고장이란 칩과 칩 사이에 존재하는 두개 이상의 넷이 연결(short)되어 발생하는 고장을 말한다. 이러한 고장을 검출하기 위한 패턴으로는 Log(N), 2Log(N), 2N등이 있으며 그중 Log(N) 패턴의 모양 및 특성은 다음과 같다.

2.3 2LOG(N) 연결 고장 검출용 패턴

2Log(N) 패턴은 Log(N)의 각 패턴에, 그 Complement 패턴을 첨가한 형태이다. 예를 들어, 테스트 대상 넷이 12개일 경우, 이들 넷 사이의 연결 고장을 검출하기 위한 2Log(N) 패턴의 길이는 8비트 길이가 된다. 이를 도시하면 아래와 같다.

N1	0	0	0	0	0	1	1	1	1
N2	0	0	0	1	1	1	1	0	0
N3	0	0	1	0	1	1	0	1	1
N4	0	0	1	1	1	0	0	1	0
N5	0	1	0	0	1	0	1	1	1
N6	0	1	0	1	1	0	0	1	0
N7	0	1	1	0	1	0	0	1	1
N8	0	1	1	1	1	0	0	0	0
N9	1	0	0	0	0	1	1	1	1
N10	1	0	0	1	0	1	1	0	1
N11	1	0	1	0	0	1	0	1	1
N12	1	0	1	1	0	1	0	0	0

표 1

2.4 2N 연결 고장 검출용 패턴

2N 패턴은 Walking 1 패턴과 Walking 0 패턴이 결합된 형태의 패턴으로 대상 넷이 8개라 가정할 때 패턴의 모양은 아래와 같다.

	Walking 1								Walking 0								
N1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
N2	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0
N3	0	0	0	0	1	0	0	0	1	1	0	1	1	1	0	0	0
N4	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	0	0
N5	0	0	0	1	0	0	0	0	1	1	0	1	1	1	0	0	0
N6	0	0	1	0	0	0	0	0	1	1	0	1	1	1	0	0	0
N7	0	1	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0
N8	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0

표 2

III. 고장 모델 및 진단 기법

본 논문에서는 이들 고장들 중 15가지에 대하여 그 고장 양상과 고장에 따른 진단 알고리즘을 제시한다[2].

3.1. 고착 고장

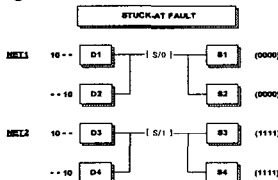


그림 1

고착 고장(stuck at fault)은 고착 고장 검출용 패턴을 인가함으로써 쉽게 검출된다.

3.2. 입력 핀 단락 고장

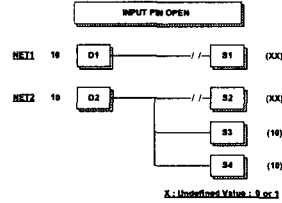


그림 2

- 1) 넷에 센스 셀이 하나만 존재할 경우 입력 핀 단락 고장은 그 고장 양상이 고착 고장과 동일하기 고착 고장으로 진단된다.(그림 2의 NET1)
- 2) 넷에 센스 셀이 하나 이상 연결되어 있을 경우, 넷에 연결된 모든 센스 셀에서 비정상패턴(all '0' 패턴 또는 all '1' 패턴)이 포착되었다면 그 패턴에 따라 S/O 고장 또는 S/I 고장으로 진단된다. 반면, 일부 센스 셀에서는 비정상 패턴이 포착되고, 일부 센스 셀에서는 정상패턴('10' 패턴)이 포착되었다면, 비정상패턴이 포착된 센스 셀이 연결된 핀에 입력 핀 단락 고장이 발생한 것으로 진단한다.(그림 2의 NET2)

3.3. 출력 핀 단락 고장

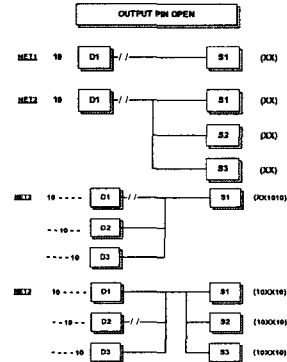


그림 3

- 1) 넷에 드라이브 셀이 하나만 존재할 경우, 출력 핀 단락 고장은 그 고장 양상이 고착 고장과 동일하기 고착 고장으로 진단된다.(그림 3의 NET1, NET2)
- 2) 넷에 드라이브 셀이 하나 이상 연결되어 있을 경우, 넷에 연결된 모든 센스 셀에 대하여 특정 Config에서 비정상 패턴이 포착되었다면, 해당 Config에서 활성화되는 드라이브 셀에 대하여 출력 핀 단락 고장으로 진단한다.(그림 3의 NET3, NET4)

3.4. 넷 단락 고장

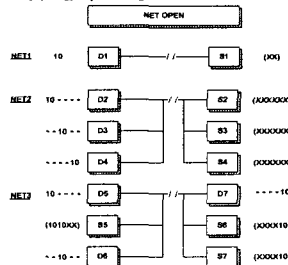


그림 4

- 1) 단일넷일 경우, 즉 넷에 드라이브 셀 하나와 센스 셀

하나만 존재할 경우, 넷 단락 고장은 그 고장 양상이 고착 고장과 동일하기 고착 고장으로 진단된다.

- 2) 넷의 한쪽에는 드라이브 셀들만 위치하고 다른 한편에는 센스 셀들만 위치할 경우, 그 고장 양상이 고착 고장과 동일하기 때문에 고착 고장으로 진단된다.
- 3) 넷의 양쪽에 드라이브 셀과 센스 셀이 고루 분포할 경우, 고착 고장이나 단락 고장이 발생하지 않았으면서도 모든 센스 셀에서 적어도 하나의 Config에서 비정상 패턴이 포착되는 경우, 넷 단락 고장이 발생했다고 진단한다.

3.5. 고착 고장 & 입력 핀 단락 고장

이 고장이 발생할 경우, 입력 핀 단락 고장이 발생한 센스 셀에 고착 고장 타입과 같은 패턴이 포착되면 고착 고장으로 진단되고, 그렇지 않을 경우 모든 센스 셀에 대하여 입력 핀 단락 고장으로 진단된다.

3.6. 고착 고장 & 출력 핀 단락 고장

이 고장이 발생한 경우, 출력 핀 단락 고장은 고착 고장에 의해 숨겨져서 검출되지 않고, 고착 고장만이 검출된다. 고착 고장을 수리한 이후에 테스트를 실시하면 출력 핀 단락 고장도 검출할 수 있다.

3.7. 고착 고장 & 넷 단락 고장

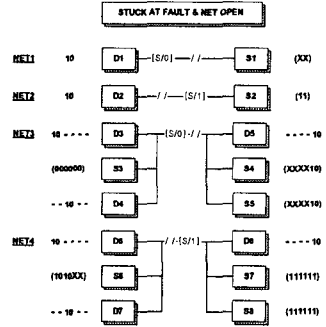


그림 5

- 1) 단일넷일 경우, 넷 단락 고장과 고착 고장의 발생 위치가 [그림5]의 NET1과 같은 경우, 실제 발생한 고착 고장의 타입과는 무관하게 센스 셀에 포착되는 패턴에 따라 S/O 고장 또는 S/1 고장으로 진단된다.
- 2) 다중넷일 경우, 고착 고장에 가깝게 위치한 센스 셀에 대하여 입력 핀 단락 고장이 발생한 것으로 진단된다. 이 고장을 수리한 이후에는 고착 고장을 검출해 낼 수 있다.

3.8. 입력 핀 단락 고장 & 출력 핀 단락 고장

한 넷에 입력 핀 단락 고장과 출력 핀 단락 고장이 동시에 발생한 경우, 입력 핀 단락 고장이 발생한 센스 셀은 비정상 패턴(all '0' 패턴 또는 all '1' 패턴)이 포착되며, 입력 핀 단락 고장이 발생하지 않은 센스 셀에는 출력 핀 단락 고장이 발생한 드라이브 셀이 활성화되는 Config에 대하여서만 비정상 패턴이 포착되고 다른 Config에서는 정상 패턴이 포착된다. 두 가지 고장 모두 검출된다.

3.9. 입력 핀 단락 고장 & 넷 단락 고장

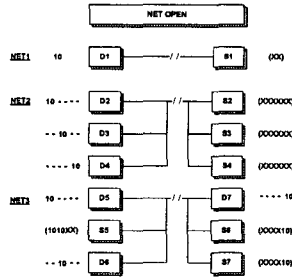


그림 6

- 1) 단일넷의 경우, 고착 고장과 그 고장 양상이 같으므로 고착 고장으로 진단된다.(그림6의 NET1)
- 2) 다중넷의 경우, 비정상 패턴이 포착된 센스 셀에 대하여 입력 핀 단락 고장으로 진단한다. 이 고장을 수리한 후에 넷 단락 고장을 진단할 수 있다.

3.10. 출력 핀 단락 고장 & 넷 단락 고장

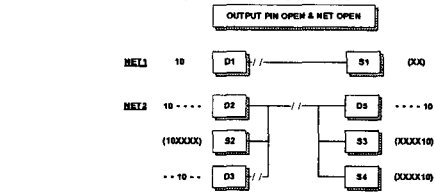


그림 7

- 1) 단일넷의 경우, 고착 고장과 그 고장 양상이 같으므로 고착 고장으로 진단된다.(그림7의 NET1)
- 2) 다중넷의 경우, 모든 센스 셀에 특정 Config에서 비정상 패턴이 포착된다. 그 Config에서 활성화되는 드라이브 셀에 대하여 출력 핀 단락 고장으로 진단한다(그림7의 NET2) 이 고장을 수리한 후에 넷 단락 고장을 진단할 수 있다.

3.11. 연결 고장

연결 고장이 발생할 경우 센스 셀에는 연결 고장이 발생한 넷들의 드라이브 셀에 주어진 패턴을 조합한 패턴이 나타나게 된다. 같은 센스 패턴을 갖는 넷들의 쌍을 찾음으로써 연결 고장을 검출할 수 있다.

3.12. 연결 고장 & 고착 고장

연결 고장과 고착 고장이 함께 발생할 경우 연결 고장의 우세성에 관계없이 고착 고장이 발생한 넷과 연결 고장이 발생한 넷들은 모두 고착 고장으로 진단된다.

3.13. 연결 고장 & 입력 핀 단락 고장

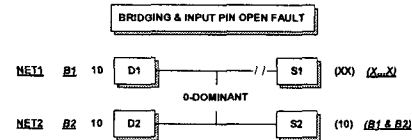


그림 8

연결 고장과 입력 핀 단락 고장이 함께 발생할 경우, 입력 핀 단락 고장이 발생한 넷의 센스 패턴은 비정상 패턴(all '0' 패턴 또는 all '1' 패턴)이 되어 고착 고장 검출용 패턴을 인가함으로써 입력 핀 단락 또는 단일넷일 경우 고착 고장으로 진단된다(그림8의 NET1).

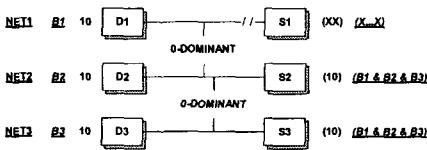


그림 9

3개 이상의 넷에서 연결 고장이 발생하고 그 넷들 중에 입력 핀 단락 고장이 발생한 넷이 포함되어 있을 경우(그림9의 경우), 입력 핀 단락 고장이 발생한 넷은 고착 고장 검출용 패턴을 인가함으로써, 입력 핀 단락 또는 단일넷일 경우 고착 고장으로 진단된다.

3.14. 연결 고장 & 출력 핀 단락 고장

이 고장의 경우, 연결 고장의 영향으로 인해 고착 고장 검출용 패턴에 의해서 출력 핀 단락 고장이 검출되지 않는 현상이 발생한다. 출력 핀 단락 고장은 연결 고장 검출용 패턴을 인가함으로써 비로소 검출된다.

3.15. 연결 고장 & 넷 단락 고장

연결 고장과 넷 단락 고장이 함께 발생하는 경우의 고장 양상은 넷 단락 고장과 연결 고장이 발생한 상대적 위치에 따라 2가 지로 나누어 볼 수 있다.

IV. 진단 알고리즘

4.1 연결 고장 진단 알고리즘

1. StuckOpenList를 입력으로 받아들인다.
2. 드라이브 패턴과 센스 패턴이 다르게 나타나는 넷들을 찾아 DifferentPatternList를 구성한다.
3. 구성된 DifferentPatternList에 속한 넷들을 조회한다.
 - 3.1 같은 센스 패턴을 갖는 넷의 쌍을 찾는다. 만약 찾았다면, BridgingList에 등록한다.
 - 3.2 StuckOpenList를 조회하여 조회한 넷에 속한 넷과 현 넷에 대하여 브리징 연산 과정을 수행하여 연결 고장 쌍을 찾는다. 만약 찾았다면 BridgingList에 등록한다.
 - 3.3 현 넷의 연결 고장 쌍을 찾을 수 없다면,
 - 3.3.1 두 넷 이상에서 연결 고장이 발생하고, 그 중 하나의 넷에서 입력 핀 단락 고장이 발생했거나, 하나의 넷은 입력 핀 단락 고장이 또 다른 하나의 넷에서 출력 핀 단락 고장이 발생한 경우에 대하여 처리한다. 만약 찾았다면 BridgingList에 등록한다.
 - 3.3.2 여전히 현 넷의 연결 고장 쌍을 찾을 수 없다면, 두 넷 이상에서 연결 고장이 발생하고, 그 중 하나의 넷이 센스 셀을 가지고 있지 않은 경우에 대하여 처리한다. 만약 찾았다면 BridgingList에 등록한다.
 - 3.3.3 여전히 현 넷의 연결 고장 쌍을 찾을 수 없다면, 두 넷 이상에서 연결 고장이 발생하고, 그 중 하나의 넷에 출력 핀 단락 고장이 발생한 경우에 대하여 처리한다. 만약 찾았다면 BridgingList에 등록한다.

V. 실험

5.1 실험 환경

제안된 알고리즘을 C++ 언어를 이용하여 Windows98에서 실행 가능하도록 구현하였다. 컴파일러는 VC++ 6.0 을 사용하였으며, 구현된 진단 시스템을 이용하여 인텔 펜티엄II - 350Mhz, 램 128M의 시스템 환경에서 아래의 실험 방법에 따라 실험을 실시하였다.

5.2 실험 방법

첫째, 고장 시뮬레이터를 통해 사용자가 원하는 고장을 고장 리스트에 주고 그 고장이 발생할 경우의 고장 결과 패턴을 얻는다. 둘째, 위와 같이 얻는 고장 결과 패턴과 인가 테스트 패턴을 제안된 알고리즘에 따라 구현된 진단 시스템의 입력으로 하여 진단 시스템을 수행시킨다. 셋째, 진단 시스템의 리포트를 통해 고장 시뮬레이터에서 발생시킨 고장이 진단되었는지를 확인한다.

5.3 실험 대상 보드

보드의 종류	PF2150	JTAGTECH	ALPGBD	MPRA60
BSD의 개수	9	15	36	2
채인의 개수	2	2	2	1
Config 개수	3	5	3	3
총 BSC의 개수	196	1448	1932	1456
2Log(N)	0	3	3	3
수행시간(단위:초)				
2N 수행시간(단위:초)	145	146	147	60

표 4

위의 표에서 알 수 있듯이 테스트 수행시간은 테스트 패턴의 종류에 따라 크게 차이가 나며, 보드 특성에 관련해서는 채인의 개수에 가장 많은 영향을 받는 것을 알 수 있다.

VI. 결론

보드의 복잡도가 점차로 커져가고 있는 시점에서 IEEE std 1149.1 표준은 보드 레벨 테스트에 대한 중요한 발판을 마련해 주었다. 본 연구의 내용인 경계스캔을 적용한 보드상의 고장 검출 기법을 이용한다면, 향후 국내 산업계에서 생산되는 보드들에 대한 신뢰성 증대 뿐 아니라 제품의 국제 경쟁력 향상에 기여할 수 있으리라 생각한다. 향후 과제로는 시스템 레벨의 테스트를 생각해 볼 수 있겠다. 현재의 보드 레벨 테스트를 확장한 개념으로, 이는 원 터치로 시스템 전체에 대한 테스트를 짧은 시간 안에 사용중인 현장에서 수행할 수 있도록 해 줄 것이다.

참고 문헌

- [1] IEEE Standard Test Access Port and Boundary Scan Architecture, IEEE Std. 1149.1-1990, IEEE Press, New York, 1990.
- [2] Harry Bleeker and Peter van den Eijnden, "Boundary-Scan Test A Practical Approach", Kluwer academic publishers, 1993.
- [3] Abu Hassan, Janusz Rajski and Vinod K.Agarwal, "Testing and Diagnosis of Interconnects using Boundary Scan Architecture", Proc. Int'l Test Conf., IEEE Computer Society Press, Los Alamitos, Calif.1988, pp.126-137
- [4] P.Wagner, "Interconnect Testing with Boundary Scan," Proc. Int'l Test Conf., IEEE Computer Society Press, Los Alamitos, Calif.1988, pp.52-57
- [5] N.Jarwala and C.Yau, "A New Framework for Analyzing Test Generation and Diagnosis Algorithms for Wiring Interconnects," Proc. Int'l Test Conf., IEEE Computer Society Press, Los Alamitos, Calif., 1989, pp.63-70
- [6] M.Abramovici, M.A.Breuer and A.D.Friedman, Digital Systems Testing and Testable Design, IEEE Press 1990.