

반도체 테스트 비용 절감을 위한 랜덤 테스트 효율성 향상 기법

이성제* · 이상석* · 안진호**

** 호서대학교 전자공학과

A Method on Improving the Efficiency of Random Testing for VLSI Test Cost Reduction

Sungjae Lee*, Sangseok Lee* and Jin-Ho Ahn**

** Hoseo University, Department of Electronic Engineering

ABSTRACT

In this paper, we propose an antirandom pattern-based test method considering power consumption to compensate for the problem that the fault coverage through random test decreases or the test time increases significantly when the DUT circuit structure is complex or large. In the proposed method, a group unit test pattern generation process and rearrangement process are added to improve the problems of long calculation time and high-power consumption, which are disadvantages of the previous antirandom test.

Key Words : Random Test, Antirandom, Low Power Test, Fault Coverage, LFSR

1. 서 론

4차 산업혁명이 태동시킨 새로운 기술, 빅데이터, 인공지능, 로봇공학, 사물인터넷, 3D프린팅, 나노바이오, 자율주행 이동체 등은 향후 국가 경쟁력과 산업을 주도할 핵심 기술로 성장하고 있다. 특히, 자율주행 자동차 분야에서 전자부품이 차지하는 비중이 증가하면서 부품 신뢰도에 대한 중요성이 점차 커지고 있다. 자동차 기능 안전성 국제 표준(ISO26262)에서는 차량 안전성 보전 등급(Automotive Safety Integrity Level: ASIL)을 안전 필수성이 가장 낮은 애플리케이션에 대한 A부터 가장 높은 D까지 정의하고 있다[1].

ISO 26262에서 정의한 기능 안전성은 “시스템 오동작에서 야기된 해저드로 인한 예측 불가능한 위험이 없음”이며 여기서 오동작은 다음 2가지 고장이 원인이 된다.

▪ 시스템 오류(Systematic Failure)

시스템 개발이나 생산, 유지 보수과정에서 발생하는 제품이나 기능의 오류

▪ 랜덤 오류(Random Failure)

무작위 결함에서 발생하는 동작 중 발생 오류로서 영구 고장(stuck-at fault)과 간헐 고장(transient fault 또는 soft error), 잠재 고장(latent fault) 등으로 구분

ASIL 등급은 A~D등급으로 분류하고 D등급으로 갈수록 고장 발생 시 더 심각한 문제를 일으킬 수 있는 중요 부품이기에 높은 신뢰성을 보장해야 한다. ASIL 등급에 따른 고장별 최대 검출 목표치는 Table 1과 같다(SPFM: Single Point Fault Metric, LFM: Latent Fault Metric).

Table 1. ASIL 등급별 고장검출률 기준

ASIL 등급	SPFM	LFM
B	≥90%	≥60%
C	≥97%	≥80%
D	≥99%	≥90%

†E-mail: jhahn@hoseo.edu

일반적으로 반도체 테스트를 위해서는 칩에 내장된 다양한 코어(또는 IP)별로 목표 고장검출률(fault coverage)을 만족시킬 수 있는 테스트 패턴 집합이 필요하고 이러한 데이터는 일반적으로 반도체 설계 회사에서 제공하고 있다. 그러나 여러 회사의 IP를 통합하는 과정이나 제조 중 발생하는 예측 불가능한 사유로 인해 기존 패턴으로 고장 검출이 불가능한 경우도 있기 때문에 어떠한 경우에도 ASIL 등급에 따른 목표 고장검출률을 달성할 수 있는 테스트 방법이 요구된다.

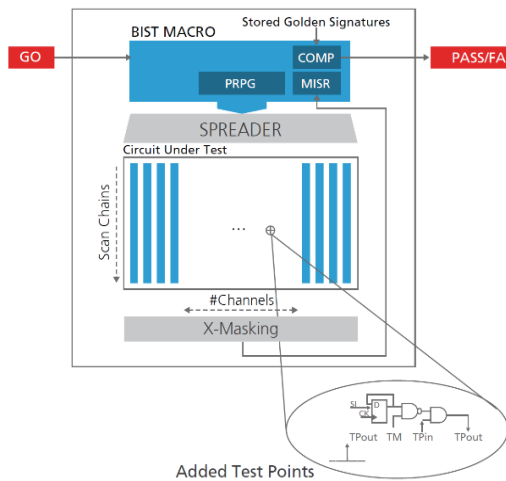


Fig. 1. Logic BIST Example[2].

Fig 1의 로직BIST(Built-In-Self-Test)[2]는 무작위적 결합에 의한 고장을 반도체 자체 테스트 과정을 통해 검출할 수 있는 DFT(Design-For-Test) 기술이며 ISO 26262 요건을 수월하게 충족시킬 수 있는 대표적인 방법이다. 그러나 차량용 반도체는 기능별 환경별 용도별로 다양하기 때문에 모든 칩에 로직BIST 기술을 적용하기는 어려운 실정이다. 반도체 테스트용 장비(Automatic Test Equipment: ATE)는 테스트 중인 웨이퍼 혹은 디바이스(Device Under Test: DUT)의 회로 구조를 알 수 없기 때문에 알고리즘(Automatic Test Pattern Generation: ATPG)에 의해서 생성된 테스트 패턴이 없다면 랜덤 테스트만 가능하다. 그러나 ATE-DUT 간 제한적인 테스트 접근성 및 테스트 인터페이스 속도 때문에 로직BIST에서 사용하는 Linear Feedback Shift Register(LFSR) 기반의 의사랜덤 테스트 방식으로는 목표 고장검출률을 달성하기까지 매우 긴 테스트 시간이 필요하여 실효성이 부족하다. 이에 로직BIST와는 다른 제한조건(테스트 접근성 ↓, 테스트 속도 ↓, 하드웨어 비용 ↓)을 고려한 랜덤 테스트 효율성 향상 방안이 필요하다.

랜덤 테스트의 효율성을 개선하기 위한 대표적인 연구 분야로 랜덤 패턴의 임의성을 개선한 반랜덤(antirandom) 패턴 생성 기법[3-8]이 있다. 하지만, 대부분의 연구가 임의성 개선 및 계산시간 축소만을 목표로 하고 있기에 반도체 테스트용 패턴 생성 기술로는 적합하지 않다.

본 논문에서는 기존 반랜덤 패턴 생성 기술을 반도체 테스트 패턴 생성 기술로 활용하기 위하여 목표 고장검출률, 테스트 시간(필요 패턴 수), 패턴 계산 시간, 패턴 소모전력량을 종합적으로 고려한 반랜덤 패턴 생성 기술을 소개한다. 제안한 방법은 반랜덤 패턴의 소모전력량 제한을 추가하여 저전력 테스트를 용이하게 하면서도 전체 고장 검출률 손실은 없거나 최소화할 수 있도록 한다.

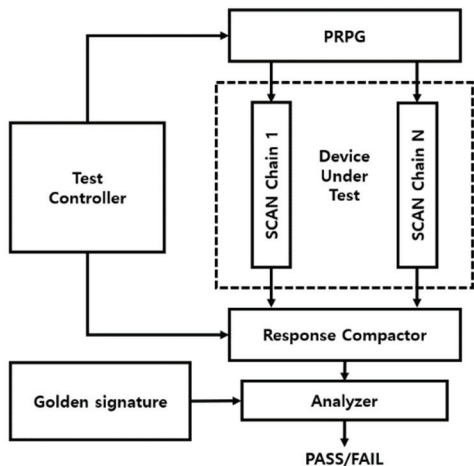
본 논문에서는 기존 반랜덤 패턴 생성 기술을 반도체 테스트 패턴 생성 기술로 활용하기 위하여 목표 고장검출률, 테스트 시간(필요 패턴 수), 패턴 계산 시간, 패턴 소모전력량을 종합적으로 고려한 반랜덤 패턴 생성 기술을 소개한다. 제안한 방법은 반랜덤 패턴의 소모전력량 제한을 추가하여 저전력 테스트를 용이하게 하면서도 전체 고장 검출률 손실은 없거나 최소화할 수 있도록 한다.

2. 기존 연구

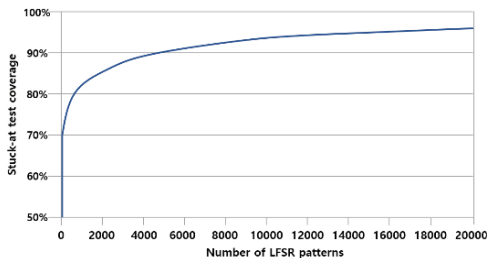
일반적인 SOC반도체는 목표 고장검출률을 달성하고자 내장형 자체 테스트(BIST)에 의한 랜덤 패턴 테스트와 고장 시뮬레이션을 통한 ATPG 패턴 테스트를 동시에 사용한다. 세부적으로 설명하면 DUT 내부 로직으로 삽입되는 PRPG(Pseudo-Random Pattern Generator)(Fig. 2(a))는 회로 구조와 상관없이 의사랜덤 테스트 패턴을 칩 동작속도로 생성하고 인가하여 초기 고장검출률을 Fig 2(b)와 같이 일정 수준 이상까지 빠르게 높일 수 있다. 이후 ATE는 ATPG를 통해서 생성한 테스트 패턴을 이용하여 목표 고장검출률(Fig 2(c))을 만족하게 된다.

로직BIST에서 사용하는 PRPG용 하드웨어는 LFSR이 주로 활용된다. LFSR은 구조가 단순하여 추가되는 하드웨어 부담없이 칩에 내장할 수 있는 장점이 있지만 주기별 반복성 및 연속 패턴의 유사성으로 인하여 랜덤 저항성 고장(random resistant fault)에 취약하고, 목표 고장검출률에 도달하기 위해서는 많은 테스트 패턴이 필요하여 결과적으로 테스트 시간이 크게 증가하게 된다.

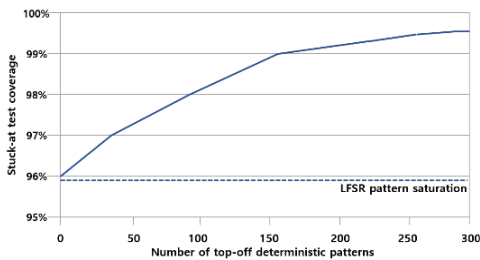
반랜덤(antirandom) 방식은 이웃하는 패턴 간 유사도를 해밍 거리(Hamming Distance) 또는 카테전 거리(Cartesian Distance) 등을 사용하여 측정하고, 그 값을 기준으로 패턴 인가 순서를 조정함으로써 고장검출률을 신속하게 높이는 방법이다[3]. 하지만, 전체 테스트 패턴에서 최대 해밍 거리 누적 값 또는 최대 카테전 거리 누적 값을 구하는



(a) SOC 반도체의 로직BIST 구조



(b) PRPG에 의한 고장검출률



(c) ATPG에 의한 고장검출률

Fig. 2. 반도체 고장검출률 달성 방법.

과정이 복잡하고 오래 걸리는 문제가 있다. 상기 문제를 해결하고자 패턴 간 거리 값을 고정시키고 테스트 패턴 순서를 스케일러블하게 조정하는 방식이 제안되었다[4]. 본 방식은 사전에 정한 거리 값에 따라 테스트 패턴 크기가 결정되므로 목표 고장검출률에 적합한 거리를 구하는 것이 중요하다. 주기적으로 입력 패턴의 비트를 스와핑하는 방식도 제안되었다[5]. 실험에서는 2ⁿ 사이클마다 각 입력 값을 스와핑하여 검출률을 향상시켰는데 스와핑 주기와 고장검출률 간의 관계가 불분명하여 실험적으로 결정해야 하는 단점이 있다. 고장검출률을 기반으로 해밍 거

리를 가변적으로 조정하는 방식(MMHD: Max-Min Hamming Distance)[6]은 짧은 주기 동안만 최대 해밍 거리를 계산하므로 계산 시간을 크게 절감할 수 있다. 한번 거리가 변경되면 해당 주기 내에는 사전에 정한 수만큼의 패턴(2-4 개 정도)만 출력하고 이를 모아서 전체 테스트를 진행한다. 상기 방식에서 주기 내 패턴 생성 수를 $2(\log_2 N + 1)$ 로 변경하는 방식[7]은 계산 시간 절감 효과는 유지하되 고장 검출률만 높이고자 하였다(N은 DUT 입력 수). 이러한 패턴 생성 주기별로 최대 카테전 거리 누적 값을 사용하여 반 랜덤 패턴을 생성하고 이를 seed로 사용하는 방식도 있다 [8]. 최대 카테전 거리 누적 값을 구하는 시간을 단축하기 위하여 누적 값 계산 방식을 데이터베이스 검색 형태로 사용하였다.

3. 제안하는 방식

기존 연구에서 확인할 수 있듯이 전체 랜덤 패턴을 반 랜덤 방식으로 구하는 것은 계산의 복잡성으로 인하여 실용성이 낮다. 또한, 반랜덤 패턴은 연속되는 패턴의 이질성을 최대화하는 것이 목표지만 이럴 경우 연속 패턴의 각 비트별 스위칭이 커지게 되어 테스트 시 소모되는 전력소모량이 늘어나 칩의 추가적인 불량을 야기할 수 있다[9]. 따라서, 랜덤 패턴의 다양성을 높이면서도 저전력 테스트가 가능한 패턴 생성 방식이 필요하다.

본 논문에서 제안한 전력소모량을 고려한 랜덤 테스트 과정을 그림 3에서 나타내었다. 전체 과정은 크게 2단계로 진행되며 각 단계별 세부 동작은 다음과 같다.

▪ (Step 1) 테스트 패턴 생성

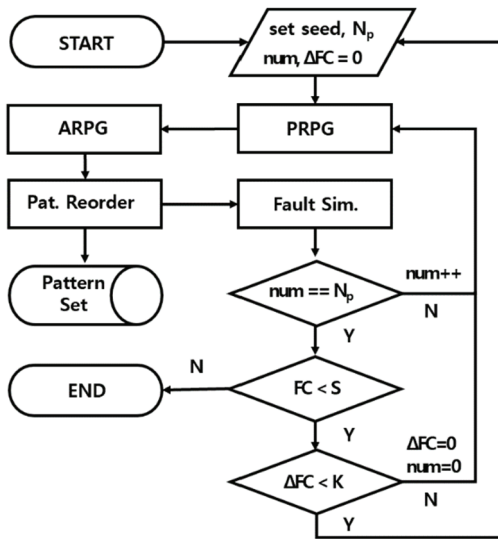
- ① 초기 seed와 seed 단위 패턴 생성 수(N_p), 목표 고장검출률(S), 고장검출률 기준 증가율(K) 설정
- ② LFSR을 이용한 의사랜덤 패턴 생성(Pseudo - Random Pattern Generation: PRPG) 및 반랜덤 패턴 생성(Anti-Random Pattern Generation: ARPG)
- ③ 상기 패턴을 저전력 테스트 용도로 변환하고자 연속 패턴 별 스위칭 수를 최소화하도록 재배열(reordering) 과정을 진행

▪ (Step 2) 고장 시뮬레이션 수행

- ④ Step 1에서 구한 패턴을 이용한 고장 시뮬레이션(Fault Simulation) 진행
- ⑤ 목표 고장검출률(S)을 만족할 때까지 생성된 패턴 수와 고장검출률을 확인: 패턴 수가 N_p 보다 작으면 ②번 과정으로 리턴. 패턴 수가 N_p 에 도달했을 때 고장검출률 증가율(ΔFC)이 기준 증가율(K)보다 작으면 seed 교체 후 다시 ②번 과정 수행

Table 2. 실험결과: Fault Coverage(%)

회로명	Pseudo-Random		Proposed Pattern		회로명	Pseudo-Random		Proposed Pattern	
	128	512	128	512		256	1024	256	1024
c432	90.5	98.1	93.9	98.7	s386	80.5	100.0	84.4	99.2
c880	91.2	97.0	93.0	97.6	s420	83.0	98.6	77.9	98.4
c1355	88.9	95.2	88.4	96.8	s713	86.8	96.1	85.9	96.5
c1908	82.0	88.6	79.9	90.9	s820	77.1	89.7	73.2	89.9
c3540	82.5	91.8	81.3	91.4	s1196	76.2	87.2	79.6	89.7
c7552	88.2	91.7	87.5	91.3	s35932	89.8	97.4	89.8	97.6



N_p : # of Patterns, num: Pattern Count, FC: Fault Coverage

Fig. 3. 전력소모량을 고려한 랜덤테스트 과정.

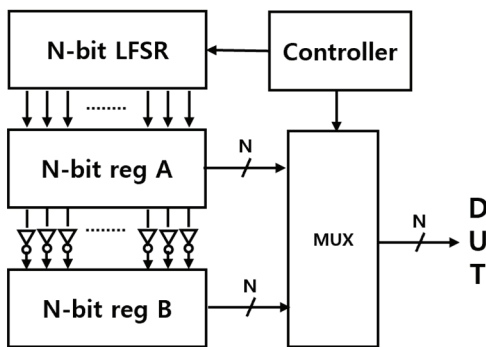


Fig. 4. 패턴 재배열을 위한 하드웨어 구조($N_p=4$).

최종적으로 생성한 테스트 패턴 집합(Pattern Set)은 ATE 또는 DIB 내장 메모리에 저장해서 사용하면 된다. 만약 패턴 집합을 모두 저장할 수 있는 공간이 없거나 실시간으로 랜덤 테스트를 하는 경우에는 N_p 의 수를 줄이고 N_p

단위로 패턴 재배열 과정을 수행하면 된다. Fig. 4는 2개의 의사랜덤 패턴과 반랜덤 패턴을 재배열하는 경우($N_p=4$)로서 2개의 레지스터와 MUX 1개를 사용하여 패턴 출력 순서를 조정할 수 있다.

4. 실험 결과

본 논문에서는 ISCAS'85 및 '89 벤치마크 회로를 대상으로 임의의 seed를 사용한 의사랜덤 패턴과 제안한 방식과의 C시뮬레이션 결과 비교를 통하여 검증하였으며, 고장 시뮬레이터는 atalanta 프로그램[10]을 사용하였다.

Table 2. 실험결과는 목표 고장검출률($S=90\%$ 이상(ASIL B 등급 기준), seed 단위 패턴 생성 수($N_p=4$)를 기준으로 일정한 수(128-1024)의 의사랜덤 패턴만 입력한 경우와 의사랜덤 패턴 및 반랜덤 패턴을 저전력 모드로 입력한 경우를 비교한 것이다. 실험에서 반랜덤 패턴은 의사랜덤 패턴의 보수로서 해밍 거리를 최대화하기 위하여 사용하였다(Fig. 4 참조).

실험 결과 제안한 방식은 초기에는 대상 회로에 따라 의사랜덤 패턴 대비 검출률이 떨어지는 경우가 종종 있으나 패턴 수가 증가할 수록 회로에 관계없이 우수한 결과를 나타내었다. 이는 반랜덤 패턴의 효과로 판단되며 반랜덤 패턴의 종류와 입력 수를 늘리면 그 효과는 더욱 커질 것으로 예상된다. 또한, 전체적인 패턴 형태는 유지하면서 N_p 단위로 패턴 입력 순서를 조정함으로써 반랜덤 패턴의 단점인 전력 소모량을 줄일 수 있었다. 회로별 전력 소모량 및 반랜덤 패턴의 종류와 전력 소모량 감소 효과에 대한 상관관계 분석은 후속 연구를 통해 제시할 계획이다.

감사의 글

이 논문은 2022년도 호서대학교의 재원으로 학술연구비 지원을 받아 수행된 연구임(202202070001).

참고문헌

1. P. Kafka, "The Automotive Standard ISO 26262, the innovative driver for enhanced safety assessment & technology for motor cars", *Procedia Engineering* 45 2 – 10, 2012.
2. Cadence, *Automotive Functional Safety Using LBIST and Other Detection Methods*, 2019.
3. S. H. Wu et al., "Antirandom Testing: A Distance-Based Approach", *VLSI Design*, vol. 2008, pp. 1-9, 2008.
4. D. B. Y. Yiunn, A. K. B. A'ain, Khor, and J. Ghee, "Scalable test pattern generation(STPG) ", *Proc. IEEE Symp. Ind. Electron. Appl. (ISIEA)*, pp. 433–435, 2010.
5. M. S. Sahari, A. K. A'ain, and I. A. Grout, "Scalable antirandom testing(SAT)", *Int. J. Innov. Sci. Mod. Eng.*, vol. 3, pp. 33–35, Mar. 2015.
6. I. Mrozek and V. N. Yarmolik, "Iterative antirandom testing", *J. Electron. Test.*, vol. 28, pp. 301–315, Jun. 2012.
7. I. Mrozek and V. Yarmolik, "Methods of synthesis of controlled random tests", *Proc. IFIP Int. Conf. Comput. Inf. Syst. Ind. Manage.*, pp. 429–440, 2016.
8. A. Alangir et al., "Multiple Controlled Antirandom Testing (MCAT) for High Fault Coverage in a Black Box Environment", *IEEE Access (Volume: 7)*, 2019.
9. K. Kim and S. Kang, "An Efficient Test Pattern Generator for Low Power BIST", *Journal of the Institute of Electronics Engineers of Korea*, vol. 47, SD, no.8, pp.29 - 35, 2010.
10. Atalanta v. 2.0, <https://github.com/hsluoyz/Atalanta>

접수일: 2023년 2월 27일, 심사일: 2023년 3월 16일,
 게재확정일: 2023년 3월 16일