

반도체 테스트 시간 단축을 위한 혼합형 랜덤 패턴 생성기

이성제* · 이상석* · 안진호*†

*† 호서대학교 전자공학과

A Hybrid Random Pattern Generator for IC Test Time Reduction

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

*† Hoseo University, Department of Electronic Engineering

ABSTRACT

Traditional pseudo-random test generation using LFSR is widely used for SOC testing due to its simplicity for hardware implementation. However, it has limitation in detecting random-resistant faults and requires a large number of test pattern leading to long test times. To overcome those problems, anti-random test has been proposed to maximize the discrepancy between successive test patterns. However, there remains a problem that it takes a long time to calculate the distance function. In this paper, a novel Hybrid Random Pattern Generation (HRPG) is presented for efficient SOC testing which combines the advantages of both pseudo-random and anti-random approaches. Experimental results demonstrate that the proposed HRPG technique can achieve over 90% fault coverage with an average of 56% fewer patterns compared to pseudo-random patterns. Hardware implementation and its simulation results are also introduced to verify the proposed method.

Key Words : LFSR, pseudo-random, anti-random, hybrid random pattern generation, soc test

1. 서 론

칩의 기능적 동작을 검사하는 Functional 테스트의 경우 테스트 대상 반도체(Device Under Test: DUT) 내 고장을 식별할 수 있는 테스트 패턴을 DUT에 입력하고 출력되는 데이터를 무고장 출력 값과 비교하여 고장 여부를 판정한다[1-2]. 이러한 테스트 패턴을 얻기 위해서는 회로 내 특정 위치에서 발생 가능한 고장을 검출할 수 있는 입력 패턴 값을 알고리즘적으로 구하는 방식과 회로 정보와 고장모델 없이 무작위로 입력 값을 결정하는 방식이 있다. 전자의 경우는 일반적으로 반도체 설계 회사에서 제공하고 있으며, 후자의 경우는 블랙박스 테스트라고 하며 가장 널리 사용하는 방식은 LFSR(Linear Feedback Shift Register)이나 CA(Cellular Automata)와 같은 로직을 사용하는 의사랜

덤 테스트이다. LFSR을 이용한 의사랜덤 패턴 생성 방식은 간단한 로직으로 구현이 가능하여 칩에 내장되는 자체 테스트(BIST) 하드웨어로 사용되어 왔다[3-4]. 그러나, 의사랜덤(pseudo-random) 형태로 만들어진 테스트 패턴은 랜덤 저항성 고장(random resistant fault)에 취약하고, 목표 고장검출률(fault coverage)에 도달하기 위해서는 많은 테스트 패턴이 필요하여 테스트 시간이 늘어나게 된다.

이러한 문제를 해결하기 위하여 반랜덤(anti-random) 패턴 방식이 제안되었다. 반랜덤 방식은 이웃하는 패턴 간 유사도를 해밍거리(hamming distance)나 카테전거리(cartesian distance)와 같은 거리 함수(distance function)를 이용하여 패턴 간 중복성을 줄여서 고장 검출의 확률을 높일 수 있는 장점이 있지만 거리를 계산하는 시간이 너무 오래 걸리는 문제가 있다[5-6]. 본 논문에서는 기존 반랜덤 테스트의 장점인 패턴의 다양성과 랜덤 테스트의 장점인 테스트 회로에 대한 유연성을 모두 제공할 수 있는 혼합형 테

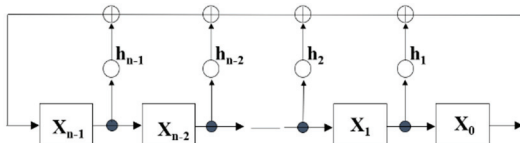
†E-mail: jhahn@hoseo.edu

스트 패턴 생성 방법과 이를 구현할 수 있는 하드웨어 구조를 소개한다.

2. 기존 연구

2.1 의사랜덤 테스트 패턴 생성

의사랜덤 패턴을 만들기 위해 사용하는 LFSR은 Fig. 1 과 같이 쉬프트 레지스터 중 일부 플립플롭의 값들이 XOR된 값을 레지스터의 입력으로 하는 피드백 루프를 추가한 형태로 구성한다. 이 때 피드백 루프에 포함된 플립플롭에 따라 생성되는 패턴의 순서와 주기가 달라지며 이것은 Fig. 1 특성 다항식(characteristic polynomial) $f(x)$ 에 의해 계산될 수 있다. 참고로 LFSR은 피드백 경로 구성 방식에 따라 XOR 피드백 회로가 쉬프트 레지스터 외부에 있는 스탠다드 방식과 쉬프트 레지스터 내부에 있는 모듈러 방식이 있으며, 패턴 생성기로 사용되는 LFSR은 일반적으로 스탠다드 방식이 사용되고 있다.



$$f(x) = 1 + h_1x + h_2x^2 + \dots + h_{n-1}x^{n-1} + x^n$$

Fig. 1. LFSR structure and characteristic polynomial.

의사랜덤 패턴은 한 주기 내 모든 수가 단 한 번씩만 출력되며, LFSR 구성에 따라 모든 수가 출력되는 것은 아니기에 90% 이상의 높은 고장검출률에 도달하기 위해서는 상당히 오랜 시간이 필요하다. 따라서, 일반적으로 일정 수준의 고장검출률을 달성하면 남은 고장을 검출하기 위하여 테스트 패턴을 알고리즘적으로 계산하는 결정론적 테스트(deterministic test)[7]로 전환하거나 LFSR 초기값 교체(reseeding)[8], 특성 다항식 변경[9] 등 다양한 방식을 사용하여 제한된 시간 내 고장검출률을 높이려는 시도를 하고 있다.

2.2 반랜덤 테스트 패턴 생성

의사랜덤의 문제를 해결하기 위하여 제안된 반랜덤 테스트 패턴 생성 방식은 이웃하는 패턴 간 유사도를 해밍 거리나 카테전거리와 같은 거리 함수를 사용하여 측정하고, 패턴 간 거리가 최대화되도록 패턴 순서를 조정하는 방식이다. Fig. 2에서 입력 크기가 PI개인 회로의 반랜덤 테스트 패턴을 구하는 전체 과정을 나타내었다.

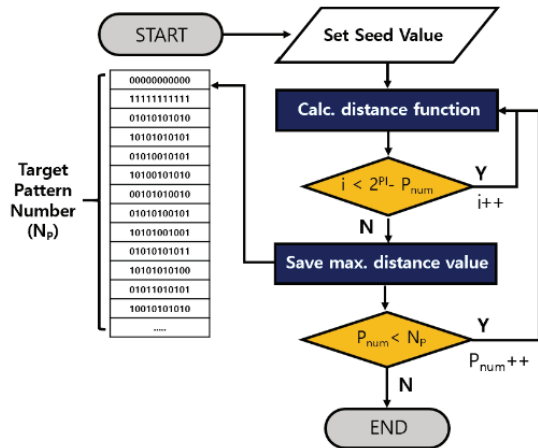


Fig. 2. Process of anti-random test pattern generation[10].

반랜덤 테스트는 패턴 간 중복성을 줄여서 고장 검출의 확률을 높일 수 있는 장점이 있지만 거리함수, 특히 카테전거리를 계산하는 시간이 너무 오래 걸리는 문제가 있다. 또한, 패턴 다양성으로 인한 고장검출률 개선 효과는 테스트 초기에 한정되며, 목표 고장검출률이 높을 경우 반랜덤 테스트 시간이 테스트 대상 회로의 구조나 입력 크기에 따라 의사랜덤 방식보다 오래 걸리는 경우도 자주 발생한다[10].

3. 혼합형 랜덤 패턴 생성기

랜덤 패턴은 보통 메모리에 저장하지 않고 하드웨어 로직을 사용하여 실속도로 생성 후 회로에 입력하고 그 결과를 분석하는 형태로 사용된다. 따라서, 랜덤 패턴 생성기는 온칩 또는 보드에 실장 가능한 하드웨어 형태로 구현 가능해야 한다. 본 논문에서는 기존 의사랜덤 테스트 방식의 장점인 유연하고 단순한 회로 구조와 반테스트 방식의 장점인 다양한 패턴 생성의 장점을 동시에 제공할 수 있는 혼합형 랜덤 패턴 생성 방법과 생성기 구조를 제안한다. 혼합형 랜덤 패턴 생성기는 패턴 간 해밍 거리를 기준으로 의사랜덤과 반랜덤 패턴을 선택적으로 생성하며 기준 값을 사용자가 임의로 조정 가능하다.

3.1 혼합형 테스트 패턴 생성 방법

해밍거리만을 기준으로 하면 패턴 간 비트 단위로 XOR 값을 구하고 해당 값을 연결(concatenation)하면 해밍 거리가 최대가 되는 패턴을 생성할 수 있다. Fig. 3은 해밍 거리가 최대가 되는 패턴을 자동으로 생성하는 방법에 대해 보여준다[10]. 상기 방식은 기존 패턴과 패턴 후보들

간의 상호 비교 없이 결정된 패턴들을 이용하여 다음 패턴을 생성하는 형태이기 때문에 P (회로의 입력 수)의 크기에 관계없이 실시간으로 패턴 그룹을 생성할 수 있다.

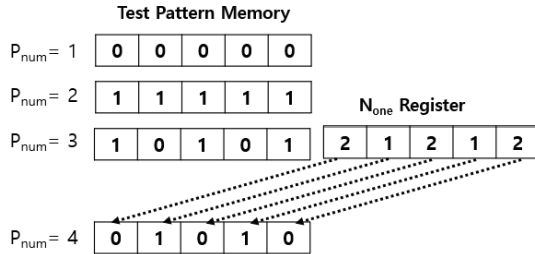


Fig. 3. Example of anti-random pattern generation using hamming distance[10].

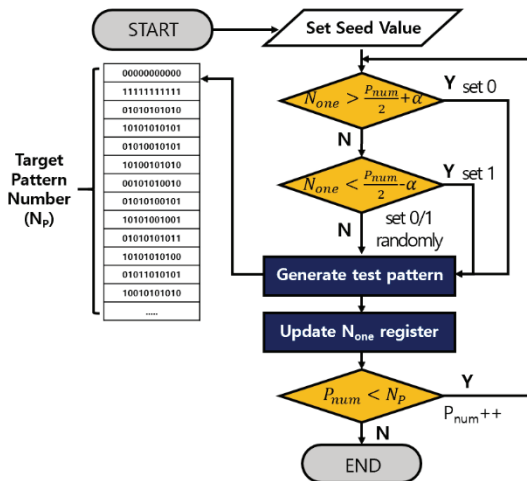


Fig. 4. Process of hybrid random test pattern generation.

혼합형 랜덤 패턴 생성기에서는 생성된 테스트 패턴들의 해밍거리를 비트 단위로 저장하는 레지스터(N_{one}) 값을 기준으로 의사랜덤 패턴과 반랜덤 패턴을 선택적으로 결정한다. 즉, N_{one} 레지스터 값이 생성된 테스트 패턴 수 (P_{num})의 50% 수준일 때 해밍거리는 최대값에 근접하게 되므로 N_{one} 레지스터 값이 $50\% \pm \alpha$ 범주 내에서는 패턴 간 다양성이 충분하다고 판단하고 의사랜덤 패턴을 생성한다. 하지만, $50\% \pm \alpha$ 범주 외에서는 패턴 간 중복성이 의심되므로 N_{one} 값이 $50\% + \alpha$ 보다 크면 0을 할당하고, $50\% - \alpha$ 보다 작을 경우 1을 할당하여 해밍거리를 $50\% \pm \alpha$ 이내로 유지하도록 테스트 패턴을 생성한다. 이때, 유효범위 α 를 크게 하면 테스트 패턴은 의사랜덤 패턴 생성 비율이 높아지며, 반대로 작게 하면 반랜덤 패턴 생성 비율이 높아지

게 된다. 유효범위 α 는 테스트 대상 회로의 특성이나 테스트 시간을 고려하여 사용자가 선택적으로 조정할 수 있다. 혼합형 랜덤 패턴 생성 방법에 대한 전체 과정은 Fig. 4와 같다.

3.2 혼합형 랜덤 패턴 생성기 구조

테스트 패턴을 하드웨어 수준에서 실시간으로 생성할 수 있는 혼합형 랜덤패턴 생성기(Hybrid Random Pattern Generator: HRPG)의 전체 구조를 Fig. 5에서 나타내었다. Fig. 5의 HRPG는 반랜덤패턴 생성기(Anti-Random Pattern Generator: ARPG)와 의사랜덤 패턴 생성기(Pseudo-Random Pattern Generator: PRPG) 출력 값을 논리 연산 후 최종 패턴으로 결정한다.

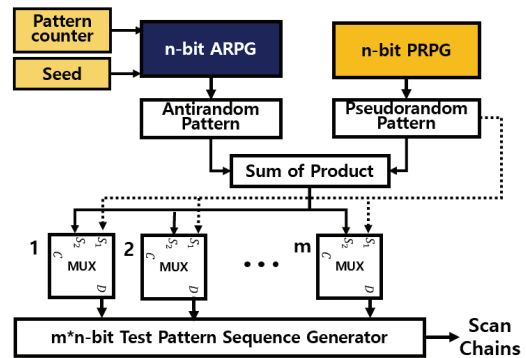


Fig. 5. Hybrid Random Pattern Generator.

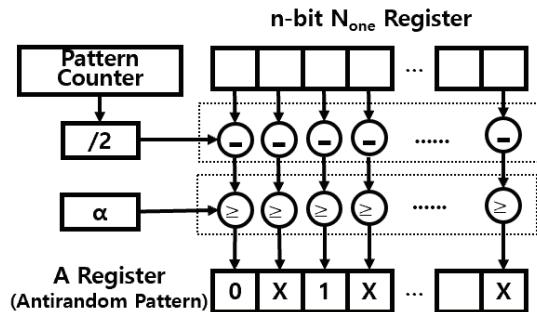


Fig. 6. Structure and operation of ARPG.

Fig. 6은 ARPG의 내부 구조 및 동작에 대한 설명이다. ARPG는 패턴 수 카운터(P_{num})와 패턴 내 1의 수를 저장하는 N_{one} 레지스터를 이용하여 다음 패턴 값의 0과 1, 그리고 무상관 값(X)을 결정한다. 각 값을 결정하는 방법은 N_{one} 레지스터 값과 P_{num} 을 2로 나눈 값과의 오차를 유효범위 α 와 비교하여 구한다. 다음 패턴을 유효범위 내에

있을 때는 X, 양의 유효범위보다 클 때는 0, 음의 유효범위보다 작을 때는 1로 설정한다. Fig. 6에서는 비트 단위 동시 계산 및 출력을 위해서 연산 로직을 비트별로 구성하는 예로 설명했지만 몇 개의 비트를 묶어 하나의 로직에서 구현하는 방식도 가능하다.

Fig. 7은 ARPG와 PRPG 출력 값의 혼합 방법을 보여준다. PRPG는 일반적으로 LFSR로 구성하며 ARPG와 PRPG에서 출력되는 패턴의 길이는 동일하다고 가정한다. Fig. 6과 같이 ARPG 연산 후 결과는 3치 형태(0/1/X)로 A 레지스터에 저장된다. 두 개의 레지스터는 한 번의 AND 연산과 OR 연산을 통하여 최종적으로 출력된다. 단, A 레지스터의 무상관 값(X)은 B 레지스터 값으로 대체하는 것으로 한다. 이 과정은 반랜덤 패턴 생성 시 발생한 무상관 값(X)을 무작위로 0 또는 1의 값을 갖게 하여 고장검출률을 높일 수 있다. 만약, 출력 패턴의 길이가 ARPG와 PRPG의 크기와 다른 경우에는 Fig. 5와 같이 MUX를 활용한 시퀀서(sequencer)를 추가하여 구현할 수 있다.

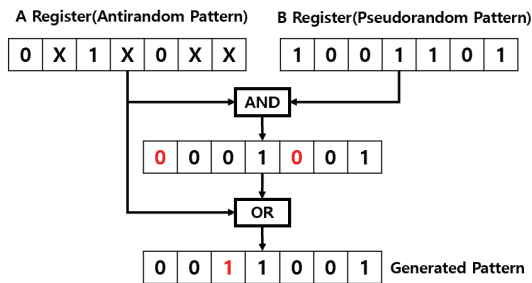


Fig. 7. Mixing method of ARPG and PRPG outputs.

4. 실험 결과

본 논문에서 제안한 방식은 ISCAS'85 및 ISCAS'89 벤치마크 회로를 대상으로 LFSR을 이용한 의사랜덤 패턴과의

고장검출률 비교를 통하여 성능을 검증하였으며, 고장 시물레이션은 Atalanta v2.0[11]을 사용하였다.

실험 결과는 Table 1에서 나타내었으며 circuit name은 실험회로명, PR은 의사랜덤 테스트, HR은 제안한 혼합형 랜덤 테스트를 가리킨다. 각 실험에서 목표 고장검출률은 90% 이상으로 지정했으며, # of pattern은 90% 이상의 고장검출률을 얻기 위해 인가된 패턴 수이고 FC는 최종적으로 달성한 고장검출률이다. HR에서 α 는 Table. 1에서 제시된 고장검출률을 얻기 위해 사용된 해밍거리 유효범위이며, 3.1장에서 설명한 바와 같이 α 가 0이면 반랜덤패턴만 생성되고 α 가 커질수록 의사랜덤패턴의 생성 비율이 높아진다. 마지막으로, 감소율은 90% 이상의 고장검출률을 얻는데 필요한 PR패턴 수와 HR 패턴 수를 비교한 값이다. 예를 들어, c432 회로의 경우 제안한 HR 방식이 PR 패턴 수 대비 45% 정도 적은 수로도 동일한 고장검출 효과를 얻을 수 있다는 것이다. Table 1의 실험 결과에서 볼 수 있듯이 제안한 HR 방식이 벤치마크 회로 특성에 관계없이 패턴 수 절감 효과가 매우 높은 것으로 확인되었다.

Fig. 8은 8비트 혼합형 랜덤 패턴 생성기(HRPG)를 HDL을 사용하여 RTL 수준으로 설계하고 검증한 결과를 보여준다. 그림의 결과는 Fig. 5 구조를 기준으로 8비트 ARPG의 초기값은 '11111111', 8비트 PRPG의 초기값은 '00000001'로 설정하고 패턴을 생성한 것이며, 그림에서 P_{num} 은 생성된 패턴 수, cnt0 ~ cnt7은 비트별로 1의 개수를 세는 N_{one} 레지스터를 나타낸다. N_{one} 레지스터와 P_{num} 의 비교를 통해 ARPG의 다음 패턴이 결정되고 Fig. 7의 ARPG와 PRPG의 혼합 방법을 통해 최종적으로 HRPG 패턴이 생성된다.

5. 결론

본 논문에서는 랜덤 테스트 패턴의 고장검출률 향상을 위한 혼합형 랜덤패턴 생성 방법과 관련 하드웨어 구조

Table 1. Fault simulation results.

Circuit Name	PR(Pseudo-Random)		Proposed Method(Hybrid Random: HR)			감소율(%)
	# of Pattern	FC	# of Pattern	FC	α	
c432	128	91.4	70	90.6	2	45.3
c5315	128	94.6	60	90.6	5	53.1
c7552	320	90.5	140	90.1	3	56.2
s1423	224	90.8	100	90.2	3	55.3
s298	128	93.5	50	93.1	2	60.9
s382	128	95.2	50	93.2	4	60.9
s400	128	95.2	50	91.9	4	60.9
s820	1504	92.9	670	90.1	12	55.4
평균 감소율(%)						56

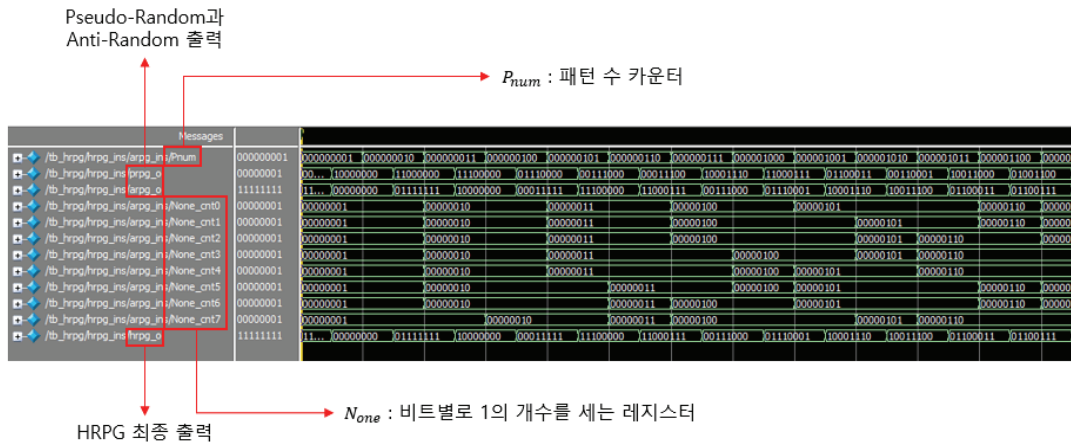


Fig. 8. RTL simulation results.

를 제안하였다. HRPG 혼합 과정 중, ARPG와 무상관 값(X)을 PRPG의 출력 값으로 대체하는 방식은 랜덤 패턴과 의사랜덤 패턴이 비트 단위로 자연스럽게 혼합되는 형태이기 때문에 의사랜덤 패턴의 상호 의존성을 줄이기 위한 추가적인 로직, 즉 위상 천이기와 같은 하드웨어를 사용하지 않아도 되는 장점이 있다. 제안한 방식은 의사랜덤덤이나 랜덤 테스트 방식[12]보다 더 짧은 시간에 더 높은 고장검출률을 제공하면서도 온칩 수준 하드웨어로 구현이 가능하여 새로운 BIST 기술로 활용이 가능하다. 또한 BIST 로직이 없거나 부족한 SOC의 경우에는 테스트 인터페이스 보드에 내장하여 ATE와 연계한 고품질 랜덤 테스트가 가능하다.

감사의 글

본 연구는 산업통상자원부(1415180306)와 KSRC 지원 사업인 미래반도체소자 원천기술개발사업(20019164)의 연구결과로 수행되었음

참고문헌

1. L-T. Wang, VLSI Test Principles and Architectures: Design for Testability, Morgan Kaufmann, 2006.
2. C. Pyton, "Scan and BIST can almost achieve test quality levels", International Test Conference, Dec. 2002.
3. B S. K. Reddy et al., "Design and Implementation of Novel FPGA Based LFSR for IOT and Smart Applications", Proc. of IEEE WIECON-ECE, Dec. 2022.
4. A. Bagalkoti et al., "Design and Implementation of 8-bit LFSR, Bit-Swapping LFSR and Weighted Random Test

- Pattern Generator: A Performance Improvement", International Conference on Intelligent Sustainable Systems, 2019.
5. S. H. Wu et al., "Antirandom Testing: A Distance-Based Approach", VLSI Design, vol. 2008, pp. 1-9, 2008.
6. T. Chen, A. Bai, A. Hajjar, A.K.A. Andrews, and C. Anderson, "Fast Anti-Random(FAR) Test Generation to Improve the Quality of Behavioral Model Verification", Journal of Electronic Testing, vol. 18, pp. 583-594, 2002.
7. M. L. Bushnell, V. D. Agrawal, Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits, Kluwer Academic, 2000.
8. B. Koenemann, "LFSR-Coded Test Pattern for Scan Designs", Proc. of European Test Conference, 1991.
9. S. Venkataraman, J. Rajski, S. Hellebrand, and S. Tarnick, "An Efficient Bist Scheme Based on Reseeding of Multiple Polynomial Linear Feedback Shift Registers", Proc. of IEEE/ACM International Conference on Computer-Aided Design, 1993.
10. S. Lee, S. Lee, and J-H. Ahn, "Method on Improving Fault Coverage of Semiconductor Test Patterns using a Distance Function", Trans. on KIEE, vol. 73, no. 1, pp. 186-191, 2024.
11. Atalanta v. 2.0, <https://github.com/hsluoyz/Atalanta>
12. S. Lee, S. Lee, and J-H. Ahn, "Method on Improving the Efficiency of Random Testing for VLSI Test Cost Reduction", Journal of the Semiconductor & Display Technology, vol. 22, no. 1, pp. 49-53, March 2023.

접수일: 2024년 10월 25일, 심사일: 2024년 11월 28일, 게재확정일: 2024년 12월 11일