

혼합 모드 BIST 테스트 패턴 생성기

Test Pattern Generator for Mixed Mode BIST

金 弘 植* · 李 抗 圭* · 姜 成 昊**

(Hong-Sik Kim · Hang-Kyu Lee · Sung-Ho Kang)



社 團 法 人 大 韓 電 氣 學 會

THE KOREAN INSTITUTE OF ELECTRICAL ENGINEERS

Test Pattern Generator for Mixed Mode BIST

金 弘 植* · 李 抗 圭* · 姜 成 昊**
(Hong-Sik Kim · Hang-Kyu Lee · Sung-Ho Kang)

Abstract - As the increasing integrity of VLSI, the BIST(Built-In Self Test) is used as an effective method to test chips. Generally the pseudo-random test pattern generation is used for BIST. But it requires lots of test patterns when there exist random resistant faults. Therefore deterministic testing is an interesting BIST technique due to the minimal number of test patterns and to its high fault coverage. However this is not applicable since the existing deterministic test pattern generators require too much area overhead despite their efficiency. Therefore we propose a mixed test scheme which applies to the circuit under test, a deterministic test sequence followed by a pseudo-random one. This scheme allows the maximum fault coverage detection to be achieved, furthermore the silicon area overhead of the mixed hardware generator can be reduced. The deterministic test generator is made with a finite state machine and a pseudo-random test generator is made with LFSR(linear feedback shift register). The results of ISCAS circuits show that the maximum fault coverage is guaranteed with small number of test set and little hardware overhead.

Key Words : BIST, 혼합 모드 테스트, FSM, 무작위 테스트, 결정 테스트

1. 서 론

오늘날 VLSI 회로를 테스트하는 데 있어서 어려운 점은 본질적으로 검침(probing)을 위해 내부 접점에 접근하기 어려운 데 기인한다. 초창기 회로들이 디지털 소자들로 구현되었을 때의 테스트는 각 접점(node)들을 검침을 사용해서 행해졌다. 그러나 요즘의 VLSI의 경우 그 내부 접점들은 그 칩의 핀을 통해서 제어되고 관측된다. 그러나 이를 하기 위해서는 많은 시간을 필요로 한다. 이러한 제약은 몇몇 접점에 테스트 접점을 삽입하고 스캔을 사용하여 테스트 모드일 때 그것들을 출력 핀에 연결함으로써 극복할 수 있다. 따라서 테스트 패턴이 주사되기도 하고 결과 값을 뽑아 비교도 할 수 있다. 그런데 이보다 더 좋은 방법은 회로 내에서 패턴을 만들기 위해 메모리 소자를 사용해서 테스트 패턴 생성기(TPG : Test Pattern Generator) 같은 추가 로직으로 묶는 것이다. 결과 분석도 출력 결과 분석기(ORA : Output Response Analyzer) 등을 통해 칩 내에서 이루어 질 수 있다. 즉 테스트 생성기에 의해 생성된 패턴을 테스트 대상 회로(CUT : Circuit Under Test)의 입력단에 가하고, 그에 대한 결과 값을 출력결과 분석기에 의해 압축한다. 여기서 압축된 결과 값을 테스트 대상 회로의 시그니처(signature)라고 한다. 이와 같은 방법을 그림 1에 나타낸 것처럼 내장된 자체 테스트라고 한다. 현재로서는 BIST가 가장 되는 회로의 복잡도와 집적도에 기인한 테스트 문제를 해결할 최적의 해결책으로 여겨지고 있다.

BIST 구현의 가장 큰 걸림돌은 패턴 생성기의 생성 시간이

길고 가격이 비싸다는 점이다. 이런 패턴 생성기는 기본적으로 세 가지로 나눌 수 있다. (의사) 전수 테스트[1], 의사 무작위 패턴[2], 그리고 결정 테스트 기법[3]이다. 전수 테스트는 모든 가능한 입력 조합을 테스트 대상 회로에 가하는 것이다. 따라서 모든 테스트 가능한 단일 고착 고장 및 다중 고착 고장이 테스트 가능하다. 이를 위해 주로 계수기(counter), 그레이 코드 생성기(gray code generator), 비선형 쉬프트 레지스터(non-linear shift register) 등이 사용된다[2]. 의사 무작위 테스트는 모든 입력 패턴 중에서 (의사) 무작위 하게 생성된 부분집합들이 테스트 대상 회로에 가해진다. 이 때 고장 검출률은 가해진 테스트 길이에 비례한다. 무작위 테스트에서 테스트 길이를 줄이고 적절한 고장 검출률을 얻기 위해 가중 무작위 테스트(weighted random test)를 사용하기도 한다. 결정 테스트 기법은 미리 만들어 놓은 패턴을 사용하는 방식이다. 그러나 오직 의사 무작위 기법만이 실용화되었다. 왜냐하면 (의사) 전수 테스트는 테스트 시간이 길고, 결정 테스트 기법은 하드웨어 오버헤드가 크기 때문이다. 의사 무작위 테스트는 테스트 대상 회로에 무작위 하게 만들어진 패턴이 가해진다. 따라서 고장 검출률은 가해진 테스트 수에 비례하게 된다. 선형 쉬프트 레지스터나 셀룰라 오토마타(cellular automata)가 테스트 생성기로 사용되는데, 일반적으로 전통적인 선형 쉬프트 레지스터(LFSR : Linear Feedback Shift Register)를 많이 사용한다. 그런데 일반적으로 BIST 패턴생성기의 가장 중요한 요건은 작은 하드웨어 오버헤드이기 때문에 주어진 회로의 효율적인 무작위 생성을 위해 LFSR/SR 구조[4]를 사용하거나, LFSR/Xor 구조[5] 등을 사용한다.

그러나 의사 무작위 방법의 가장 큰 단점은 높은 고장 검출률을 보장하기 힘들다는 것이다. 이 점을 극복하기 위해 제안된 방법은 다음과 같다. ROM을 사용한 방법[6, 7, 8], 셀룰라

* 準 會 員 : 延世大 大學院 電氣工學科 碩士課程
** 正 會 員 : 延世大 工大 電氣工學科 助教授 · 工博
接受日字 : 1998年 5月 8日
最終完了 : 1998年 6月 15日

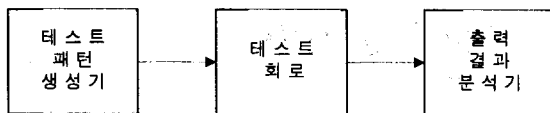


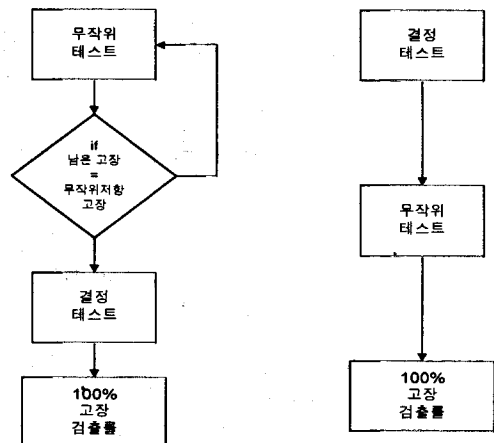
그림 1 내장된 자체 테스트 구조
Fig. 1 BIST architecture

오토마타(Cellular Automata)[9], 비동기 계수기(asynchronous counter)[10], 선형 및 비선형 피드백 쉬프트 레지스터(Linear and Non-Linear Feedback Shift Register/LFSR, NLFSR)[11, 12, 13]와 계수기와 디코더를 이용한 방법[14] 등이 있다. 그러나 높은 고장 검출률을 보장하기 위해서는 결정 테스트 기법을 사용하는 것이 가장 확실하다. 앞서 설명하였듯이, 이것은 자동 테스트 생성(ATPG : Automatic Test Pattern Generation)으로 만든 테스트를 저장해 두었다가 테스트 모드에서 사용하는 방법인데 주로 ROM을 사용해서 구현된다. 이 때 저장되는 테스트 집합은 무작위 테스트에 비해 매우 작기 때문에 테스트 시간을 상당히 단축시킬 수 있다[15]. 그러나 ROM에 결정 패턴을 저장해 두었다가 패턴을 가하는 것은 너무 큰 하드웨어 오버헤드를 필요로 한다. 반면에 오버헤드가 상대적으로 낮은 것들, 예를 들면 LFSR/NLFSR은 무작위 저항고장을 잡기 어려운 단점이 있다. 따라서 본 논문에서 채택하는 방법은 혼합 테스트 구조이다. 이것은 의사 무작위 테스트를 수행하고 그것으로 잡아내기 어려운 고장을 결정 테스트로 해결하는 방식이다. 즉 혼합 테스트는 기본적으로 두 부분으로 나뉜다. 의사 무작위 테스트와 결정 테스트인데, 의사 테스트의 결과 무작위 저항 고장만이 남을 경우 결정 테스트를 수행하여 100% 고장 검출률을 얻어낸다. 결정 테스트 생성기를 설계하기 위해 FSM(finite state machine) 구조를 채택하였다. 그런데 모든 테스트를 FSM만으로 결정 테스트를 하는 것은 오버헤드가 너무 크기 때문에 무작위 저항 고장에 대한 패턴만을 FSM을 이용해서 가해주고 나머지 고장은 의사 무작위 패턴생성으로 잡아주는 방식을 채택하였다. 그리고 결정 테스트를 구할 때에는 테스트 수와 오버헤드 사이의 트레이드 오프(trade-off)를 감안해서 결정 패턴 수를 결정하였다.

혼합 테스트 생성기의 기본 개념이 II절에서 설명되고 III절에서는 혼합 테스트를 위해 고안한 하드웨어 구조를 설명한다. IV에서는 혼합 테스트의 길이와 면적 사이의 관계를 논의하고 본 논문에서 사용한 최적의 발견법을 설명한다. V절에서는 실험결과, 즉 고장 검출률과 추가 하드웨어 오버헤드를 분석한다. 마지막으로 VI에서 결론을 맺는다.

2. 혼합 테스트의 방법론

일반적으로 의사 무작위 테스트만으로는 만족스런 결과를 얻기가 매우 어렵다. 왜냐 하면 의사 무작위 테스트의 경우 고장 검출률은 테스트 패턴 수에 비례하기 때문이다. 또한 목적 고장 검출률을 가정하고 수학적인 모델을 사용하여, 테스트 길이를 구한다고 해도 무작위 패턴 저항 고장이 존재할 경우 높은 고장 검출률을 만족시키는 것이 불가능해질 수 있는 단점이 있다. 즉, 이러한 무작위 저항 고장은 고장 검출확률에 지대한 영향을 미쳐서, 최종 테스트 길이를 비실용적으로 길어지게 만든다. 더욱이 보다 실제적이고 복잡한 고장 모델인 지연고장이나



(a) 자동 테스트 생성 (b) 혼합 테스트

그림 2 자동 테스트 생성과 혼합 테스트의 순서도

Fig. 2 The flow chart of ATPG and mixed test

다중 고장 등을 테스트하고자 할 경우, 단일 고장고장을 잡기 위한 의사 무작위 테스트 패턴은 더 이상 쓸모가 없다. 결론적으로 오늘날 실제 회로 고장의 복잡성이 증가되는 현실을 고려할 때, 오로지 의사 무작위 패턴만을 사용하는 것은 양질의 고장 검출률을 제공하는 데는 부적절하다고 할 수 있다. 따라서 본 논문은 결정 테스트의 장점을 응용한 혼합 테스트 방식을 도입한다.

혼합 테스트 생성의 알고리즘은 일반적인 자동 테스트 생성의 것과 거의 같다. 우선 유사 무작위 테스트를 수행하여, L_p 개의 무작위 패턴을 만들어 고장 시뮬레이션을 수행한다. 그 결과 적절한 테스트 길이로 100% 고장 검출률을 얻을 수 있다면 무작위 테스트로 완료한다. 그런데 CUT에 무작위 저항 고장이 남아있는 경우에는 그 고장에 대해서 자동 테스트 생성을 수행하여 L_d 개의 결정 테스트를 구한다. 총 $L_p + L_d$ 개의 테스트로 최대의 고장 검출률을 얻을 수 있다.

이렇게 구한 테스트를 가지고 실제로 혼합 모드 테스트하는 것은 자동 테스트 생성의 순서와 반대로 한다. 왜냐 하면 무작위 테스트를 하고 결정 테스트를 수행하면 반드시 $L_p + L_d$ 개의 테스트를 가했을 때, 최대의 고장 검출률을 얻을 수 있다. 그러나 그 순서를 바꾼다면, 동일한 고장검출률을 $L_p + L_d$ 개 이하의 테스트로 보장할 수 있기 때문이다. 즉 $L_p + L_d$ 개미만의 테스트로 최대의 고장 검출률을 얻을 수 있는 것이다. 이런 혼합 모드 테스트의 순서도를 자동 테스트 생성의 것과 비교해서 그림 2에 그려 놓았다. 실제로 결정 테스트를 수행하는 것은 테스트 생성 단계에서 얻은 결정 테스트를 저장해 두었다가 테스트 대상 회로에 가함으로써 이루어진다. 기본적으로 결정 패턴 생성기의 목적은 자동 패턴생성기로 얻은 결정 패턴을 그대로 CUT에 가하는 것이기 때문에 무작위 저항 고장에 대해 높은 검출률을 보장할 수 있다. 일반적으로 결정 테스트를 위한 테스트 패턴은 대개의 경우 ROM에 저장시켜 놓았다가 사용하는 방식이 쓰인다. 그러나 BIST의 경우 ROM은 그 하드웨어 오버헤드가 감당하기에는 너무 크기 때문에 본 논문에서는 FSM을 사용하여 결정 테스트 생성을 시도한다. 그림 3은 FSM을 사용한 결정 테스트 생성기 구조이다. 저장된 테스트를 모두 사용한 뒤에는, 의사 무작위 테스트를 시행하게 된다.

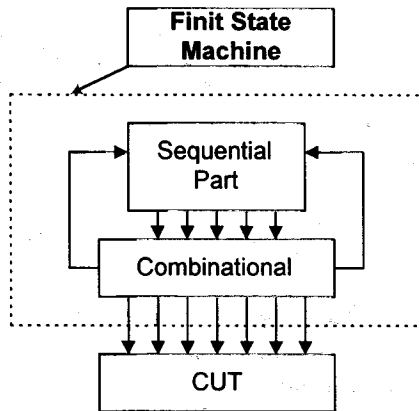


그림 3 FSM을 이용한 결정 패턴 생성기
Fig. 3 Deterministic pattern generator with FSM

3. 혼합 테스트 패턴 생성기의 구조

최대의 고장 검출률을 유지하면서 오버헤드를 줄일 수 있는 방법은 결정 테스트와 의사 무작위 테스트를 혼용하는 혼합 테스트를 사용하는 것이다. 기본적으로 본 논문에서 채택하는 것은 의사 무작위 패턴은 LFSR/SR로 결정 테스트는 FSM으로 만들어서 가하는 것이다. 의사 무작위 테스트는 그림 4에 나타난 16 비트 LFSR과 쉬프트 레지스터를 사용해서 생성하였다. 즉 16 비트 LFSR로 의사 무작위 패턴을 생성하고 쉬프트 레지스터를 사용하여, CUT의 입력 개수에 맞는 패턴으로 확장하였다.

그림 5는 n 비트의 입력을 가진 테스트 대상 회로를 테스트 하기 위한 혼합 테스트 생성기의 구조이다. 혼합 테스트는 의사 무작위 테스트와 결정 테스트의 두 부분으로 나뉘기 때문에, 의사 무작위 테스트에서 정해진 시간에 결정 테스트로 넘어갈 수 있는 제어 로직(control logic)이 필요하다. 그리고 그림 5의 구조에서는 총 오버헤드를 줄이기 위해 FSM과 LFSR이 폴립플롭을 공유하는 형태를 도입했다. 따라서 생성기의 순차 논리 회로(sequential part)는 두 부분으로 분할된 구조를 갖는다. 이것이 Sequential part 1과 Sequential part 2이다. 이것들은 제어 로직의 sel 신호에 따라서 연결되어 LFSR이 되기도 하고, 분리되어 FSM이 되기도 한다. 이중 Sequential part 2는 FSM을 위한 부분인데, 결정 테스트의 개수에 따라 그것의 상태를 생성할 수 있는 m 개의 폴립플롭으로 구성된다. 결정 테스트 단계에서 Sequential part 2와 조합부분(combination part)이 합쳐져서 생성된 FSM은 n 비트의 결정 테스트를 만들어 낸다.

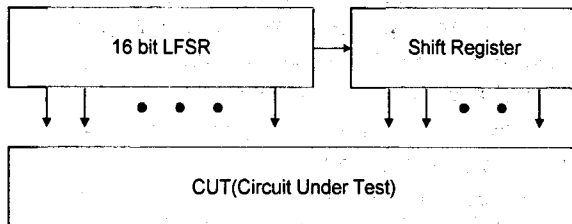


그림 4 무작위 테스트 생성을 위한 LFSR/SR 구조
Fig. 4 LFSR/SR architecture for pseudo-random test generation

다음은 본 회로의 동작을 기술한 것이다.

- 1) 테스트신호가 들어오면 제어로직은 순차부분(Sequential part)을 분할하고 조합부분을 Sequential part2에 연결해서 FSM을 구성하여 결정 테스트를 수행한다.
- 2) 결정 테스트를 모두 수행한 뒤 제어로직은 Sequential part1, Sequential part2와 XOR 부분을 연결해서 LFSR/SR을 구성한다.
- 3) 최대의 고장 검출률을 얻을 때까지 LFSR에 의해 의사 무작위 테스트를 수행한다.

그림 6은 제어 로직의 회로도이다. 여기서 의사 무작위 테스트와 결정 테스트를 구분해 주기 위한 신호는 마지막 무작위 테스트 값에 따라 0 또는 1에 연결해 놓은 n 비트 입력과 실제 n 비트 FSM의 출력 단의 값을 비교해서 생성한다. 이것은 FSM이 생성하는 마지막 결정 테스트 값이 테스트 대상 회로에 가해진 후에는 혼합 테스트 모드가 바뀌어야 하기 때문이다. 즉 결정 테스트가 모두 끝난 뒤에 무작위 테스트를 계속 수행하기 위해 sel 신호를 변화시켜 주는 것이다.

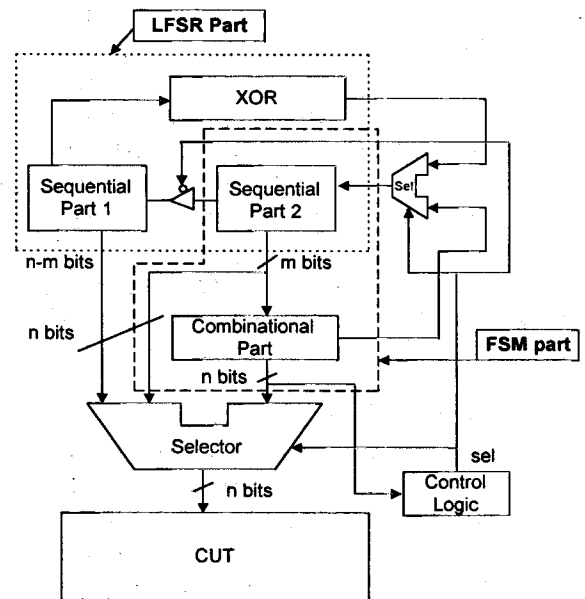


그림 5 FSM을 이용한 혼합 테스트 생성기 구조
Fig. 5 Mixed test generator with FSM

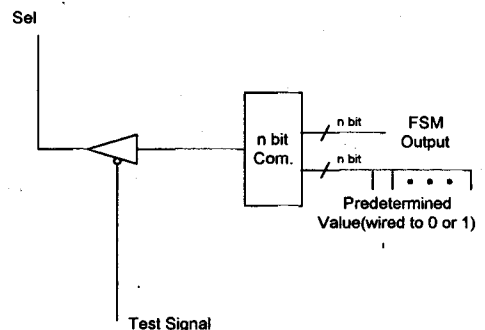


그림 6 제어 회로도
Fig. 6 Control logic diagram

4. 혼합 테스트

최근 BIST를 구현하는데 있어 고려해야 할 점은 효율성, 스피드, 경제성 등이다. 즉 높은 고장 검출률을 보장하면서 적은 비용으로 빠른 테스트를 구현할 수 있어야 한다는 것이다. 이를 위해서 제안하는 것이 혼합 테스트 기법인데, 이것은 결정 테스트에 이어 의사 무작위 테스트가 연속적으로 계속된다. 따라서 혼합 테스트 기법은 높은 고장 검출률을 유지하면서 전체 테스트 시간을 단축시키는 장점을 가지게 된다. 그런데 일반적으로 테스트 생성 시, 어느 정도의 의사 무작위 패턴을 가하고 결정 테스트를 구하는 것이 가장 효율적인지는 알려지지 않은 것이 사실이다. 이 점은 생성기의 하드웨어 오버헤드와 테스트 시간과의 관계와 밀접한 관계를 가지기 때문에 본질적으로 매우 중요하다. 이 문제를 보다 깊이 이해하기 위해 우선 의사 무작위 테스트(Lp)를 가하여 고장 시뮬레이션을 수행한다.

ISCAS c5315 회로에 관한 무작위 테스트 고장 시뮬레이션 결과가 그림 7에 나타나 있다. c5315는 총 5291의 고장이 존재하는데, 이 중 59개는 무해고장이기 때문에 모두 5232개의 고장만을 고려하여 고장 검출률을 산출하기로 한다. 256개의 의사 무작위 패턴으로 약 94.78%까지 빠르게 수렴한다. 약 512개 이후의 테스트는 그 효율성이 급격히 떨어짐을 알 수 있다. 또한 1000이상의 패턴을 가해도 100% 고장 검출률을 얻기는 어렵다. 즉, 의사 무작위 테스트의 효율성은 초반부에만 보장된다는 것이다. 후반부로 갈수록 효율성은 급격히 떨어지고, 결국 5440 개의 패턴을 가해야만 100% 고장검출이 가능함을 알 수 있다.

혼합 테스트에서는 Lp 개의 의사 테스트와 Ld 개의 결정 테스트가 필요하다. 따라서 최대 Lp + Ld 개의 테스트가 가해진다. 이와 같이 Lp + Ld 개의 테스트에 의한 c5315의 시뮬레이션 결과가 그림 8에 있다.

만일 256개의 의사 무작위 패턴이 가해졌을 때, 66개의 결정 패턴만을 가하면 고장 검출률 100%를 보장할 수 있다. 따라서 총 322개의 패턴으로 고장 검출률 100%를 얻을 수 있는 것이다. 그리고 의사 무작위 패턴을 더 많이 가하면 총 테스트 수는 길어지지만 Ld 값을 줄일 수 있다. 총 5440개의 테스트로 100%를 얻을 수 있는 무작위 테스트보다 테스트시간을 상당히 단축시킬 수 있는 것이다. 그런데 결정 테스트의 길이(Ld)는 하드웨어 오버헤드와 직접적으로 연관되기 때문에, 우리는 테스트 시간과 면적 오버헤드 사이의 트레이드 오프를 고려해야

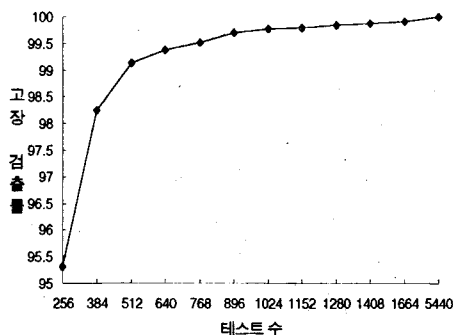


그림 7 c5315 고장 검출률 대 의사무작위 패턴 수
Fig. 7 Fault coverage of c5315 vs the number of pseudo-random test patterns.

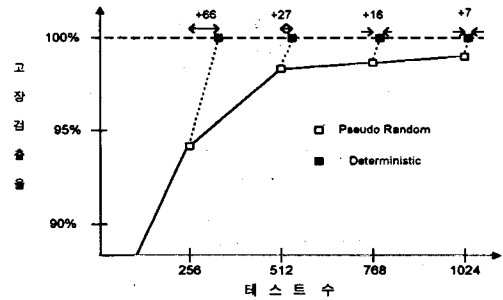


그림 8 c5315 고장 검출률 대 혼합 테스트 수
Fig. 8 Fault coverage of c5315 vs the number of mixed test patterns

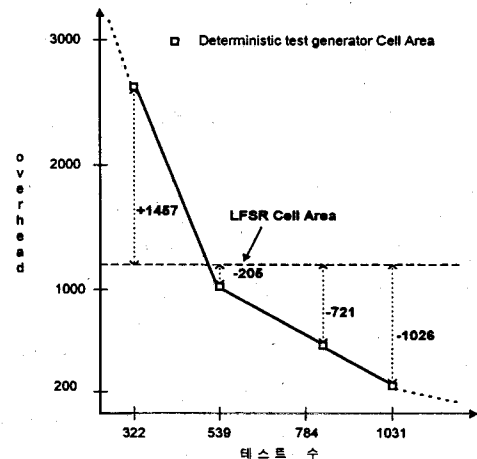


그림 9 c5315 결정 패턴 생성기 셀 면적 대 테스트 수
Fig. 9 Cell area of c5315 vs the number of total tests

만 한다. c5315 회로의 테스트 길이에 관한 오버헤드 증가 관계에 대한 실험결과가 그림 9, 10에 있다. 그림 8, 9, 10은 의사 무작위 테스트를 많이 할수록 사용할 결정 패턴은 줄어든다는 사실을 잘 보여준다. 다시 말하면, 테스트 시간이 증가하는 대신 오버헤드는 준다는 것이다. 그림 8에서 의사 무작위 패턴 (256개)을 가하면 고장 검출률 94.78%를 얻을 수 있었다. 이때 64.87%의 면적을 추가로 삽입하여 결정 테스트를 수행하면 66개의 패턴만을 추가하면, 총 322개의 테스트로 100% 고장 검출률을 얻을 수 있었다(그림 9). 그런데 1024개의 의사 무작위 테스트를 수행한 후 결정 테스트로 넘어가면 총 1031개로 100% 고장 검출률을 얻을 수 있었는데, 이를 위해 5.57%의 결정 테스트 생성기만을 추가해주면 된다. 즉, 약 3배의 테스트 시간이 걸리는 대신 오버헤드 추가는 약 1/20로 줄게 된다. 결론적으로 어느 순간에서 결정 테스트로 넘어가야 하는 가하는 문제는 시간과 면적의 함수를 고려해야 하는 것이다.

본 논문의 경우는 다음과 같은 방법을 사용하기로 한다. 우선 테스트 생성 시 의사 무작위 테스트에서 결정 테스트로 넘어가는 순간은, 무작위 테스트의 효율성이 실용적이지 못할 정도로 낮아진 때라고 볼 수 있다. 다시 말하면 아직까지 무작위 테스트가 효율성을 가진 때에는 무작위 테스트를 사용하는 것이 필요하다는 것을 의미한다. 즉 의사 무작위 테스트 저항 고장에 대해서만 결정 테스트를 하는 것이 합리적이다. 우리는

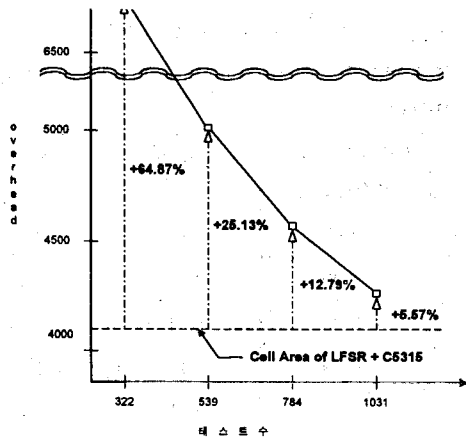


그림 10 c5315의 결정 테스트 생성기 면적 증가율

Fig. 10 The area increasing rate of deterministic test generator of c5315

이 의사 무작위 테스트 저항 고장을 다음과 같이 정의한다. 우선 LFSR/SR을 사용하여 무작위 패턴을 얻어 고장 시뮬레이션을 수행한다. 이때 우리의 목적은 이것을 통해 높은 고장 검출률의 테스트 길이를 구하려는 것은 아니라, 다만 고장 시뮬레이션으로 무작위 저항 고장에 대한 테스트 패턴을 구하려는 것이다. 따라서 320개의 패턴이 연속적으로 고장을 검출하지 못한 순간 이후의 고장을 무작위 저항 고장으로 정의하고, 그 때까지를 의사 무작위 테스트 길이로 정하였다. 즉, 그 순간에서 중단하고 그때까지 잡아내지 못한 고장에 대해 자동 테스트 패턴 생성을 해서 결정 테스트를 구하였다. 실제로 이와 같은 방법으로 구한 결정 패턴은 양질의 테스트 패턴이라고 할 수 있다. 만일 이 경우 모든 패턴을 잡아낸 경우나 위의 방법을 사용했을 때 무작위 테스트가 비실용적으로 길 경우에는, 남은 고장 개수가 급격히 줄어드는 점을 결정 테스트로 넘어가는 순간으로 정했다. 이런 식으로 결정 테스트를 구한 뒤에, 테스트 생성기를 합성한다. 이런 테스트 생성기는 결정 테스트를 수행하고, 나머지 고장들을 무작위 테스트를 통해 검출한다.

5. 실험결과

전 장과 이번 장에서 언급될 하드웨어 오버헤드에 관한 값들은 Synopsys Design Analyzer로 구현된 셀 면적들이다. LFSR과 ISCAS 벤치 회로들의 하드웨어는 먼저 VHDL 언어로 구현하여 Synopsys Design Analyzer로 자동 합성하였고, 결정 테스트를 위한 FSM 부분은 SIS 1.2로 설계하여 Verilog로 변환한 뒤 Synopsys Design Analyzer로 합성하였다. 표 1은 각 회로의 LFSR/SR에 의한 무작위 테스트의 결과이다. 여기서 모든 회로들은 전주사(full scan)를 가정하였고, 16 비트 LFSR/SR을 무작위 테스트 생성기로 사용하였다. 1 열의 값들은 각 회로의 총 고장, 테스트 가능고장 및 무해고장의 수를 의미한다. 각 고장들은 자동 테스트 생성기로 계산하였으며 역추적(backtrack) 수는 10000으로 제한하였다. 2열은 무작위 테스트의 수와 테스트 결과 남은 고장의 수이다. 이 고장들을 우리는 무작위 저항 고장으로 간주하여 자동테스트 생성기를 통해 결정 테스트를 구하였고 그 개수가 4 열에 명시되어 있다.

표 2는 앞에서 구한 결정 테스트 패턴으로 FSM을 구현하여

표 1 ISCAS 회로들에 대한 고장 시뮬레이션 결과

Table 1 The fault simulation results of ISCAS circuits

		고장 수			무작위 테스트		결정 테스트 수
		총고장수	테스트 가능 고장 수	무해 고장	테스트 수	남은 고장 수	
조합 회로	c432	524	520	4	256	7	6
	c499	758	750	8	288	8	8
	c880	942	942	0	736	24	15
	c1355	1574	1566	8	2016	33	10
	c1908	1870	1861	9	3008	19	14
	c2670	2747	2630	117	3552	226	57
	c3540	3290	3153	137	2240	17	13
	c5315	5291	5232	59	1120	9	8
순차 회로	c6288	7744	7710	34	96	20	5
	s444	474	460	14	224	6	4
	s510	568	568	0	288	9	5
	s526	555	554	1	2752	17	11
	s641	467	467	0	2528	10	10
	s713	543	515	38	2528	10	10
	s820	850	850	0	4240	30	15
	s832	870	856	14	4640	33	13
	s838	857	857	0	1568	104	56
	s953	1079	1079	0	5568	16	9
	s1196	1242	1242	0	5824	22	14
	s1238	1355	1286	69	6208	22	14
	s1423	1515	1501	14	1440	24	15
	s1488	1486	1486	0	2048	6	5
	s1494	1506	1494	12	2048	6	5

그것으로 결정 테스트를 수행한 결과이다. 1열은 ISCAS들을 의사 무작위 테스트하기 위한 BIST 회로의 면적이다. 여기서 결과 압축을 위한 회로는 고려하지 않았다. 2열은 결정테스트 생성기와 LFSR이 공유하는 풀립플롭의 셀면적이고 3열은 결정 테스트를 위한 추가 면적이다. 대부분의 회로들에 대해 결정 테스트 생성기의 면적은 상당히 적은 수준을 유지했는데, c2670이나 c880 같은 몇 개의 회로들은 그렇지 못하였다. 이것은 이들 회로들의 무작위 저항 고장의 수가 과대함을 의미하는데, 이것들은 무작위 테스트 길이를 증가한다해도 좀처럼 줄어들지 않는 속성을 지니고 있다. 따라서 이 회로들에서 결정 테스트 생성기의 면적을 줄이기 위해서는 설계 단계에서 테스트를 고려하는 DFT(design for testability) 기법을 사용하거나, 무작위 테스트의 성능을 향상시키기 위해 가중 무작위 테스트 등의 방법이 적용되어야 할 것이다.

표 3은 면적 증가 분에 대한 어느 정도의 성능 향상이 이뤄졌는지를 보여준다. 여기서 면적 증가 분은 의사 무작위 테스트

표 2 합성 결과

Table 2 Synthesis results

		테스트회로와 무작위 테스트회로의 면적	공유 면적	결정 테스트를 위한 추가면적
조 합 회 로	c432	649	21	32
	c499	962	21	22
	c880	1168	28	187
	c1355	1275	28	39
	c1908	1203	28	113
	c2670	3629	42	2691
	c3540	2611	28	119
	c5315	4653	21	140
	c6288	3085	21	17
순 차 회 로	s444	549	14	10
	s510	624	21	16
	s526	618	28	55
	s641	1093	28	93
	s713	1115	28	94
	s820	757	28	90
	s832	761	28	70
	s838	776	42	796
	s953	1010	28	79
	s1196	1161	28	104
	s1238	1167	28	122
	s1423	1935	28	259
	s1488	1454	21	13
	s1494	1452	21	15

트 생성기와 테스트 대상회로를 합한 면적에 대해 결정 테스트를 추가하기 위한 면적 증가 분을 의미한다. 대부분의 ISCAS 회로에서 적은 면적의 추가에 의해 100% 고장 검출률을 보장할 수 있었다. 즉 무작위 테스트만을 수행하는 것보다 더 나은 결과를 얻을 수 있었다. 예를 들면 c1355 회로의 경우 무작위 테스트만을 하는 회로에 3.61%의 FSM을 추가하여 단지 10개의 테스트 증가(총 2026개의 테스트)로 98.98%의 고장 검출률에서 100%의 고장 검출률을 얻을 수 있었다. FSM 추가분의 증가량이 많은 것은 c880과 같이 본회로의 크기가 극히 작은 경우와 c2670과 같이 무작위 테스트 저항 고장이 많은 경우였다. 결정 테스트의 개수가 커서 테스트 생성기의 면적이 커지는 현상은 회로의 입력 개수가 회로 크기에 비해 큰 것도 한 원인인데, 이것을 극복하기 위해서는 앞서 언급하였듯이 무작위 테스트 생성기의 성능을 향상시키거나, 무작위 테스트를 상대적으로 증가시켜 결정 테스트 부분을 감소시키는 방법 두 가지를 생각할 수 있다. 전자의 경우는 가중 무작위 기법이나 초기값 재변경(resceding) 등의 방법을 생각할 수 있다. 후자의

표 3 ISCAS 회로의 혼합 테스트를 위한 면적 증가량

Table 3 Increasing area rate of ISCAS circuits for mixed test

		무작위 테스트의 고장검출률	% 면적 증가율	혼합테스트의 고장검출률
조 합 회 로	c432	98.65	4.93	100
	c499	98.40	2.29	100
	c880	97.45	16.01	100
	c1355	97.89	3.06	100
	c1908	98.98	9.39	100
	c2670	91.41	74.15	100
	c3540	99.46	4.56	100
	c5315	98.71	3.00	100
	c6288	99.74	0.55	100
순 차 회 로	s444	98.70	1.82	100
	s510	98.42	2.56	100
	s526	96.93	8.90	100
	s641	97.86	8.51	100
	s713	98.06	8.43	100
	s820	96.47	11.89	100
	s832	96.14	9.20	100
	s838	87.86	102.58	100
	s953	96.39	7.82	100
	s1196	98.39	8.96	100
	s1238	98.29	10.45	100
	s1423	98.40	13.39	100
	s1488	99.60	0.89	100
	s1494	99.60	1.03	100

방법은 일반적으로 불가능하다. 왜냐하면 결정 테스트가 긴 경우는 무작위 저항 고장이 상당히 많을 때인데, 이 때는 무작위 테스트로 잡아내기 어렵기 때문이다. 3열은 혼합 테스트를 수행한 고장 검출률이다. 이 결과는 혼합 테스트에 의해 최대의 고장 검출률을 얻었음을 보여 준다. 일반적으로 무작위 테스트에 의해 고장검출률을 끌어올리는 것은 매우 어렵다. 따라서 혼합 테스트를 사용함으로써 적은 패턴의 추가만으로 최대의 고장 검출률을 얻었다는 것은 매우 고무적인 결과라고 할 수 있다. 이것은 표 4를 보면 더욱 분명하게 알 수 있다.

표 4는 기존의 의사 무작위 기법과 FSM을 이용한 혼합테스트 기법의 테스트 시간과 고장 검출률과의 관계를 비교해 놓은 것이다. 표에서 알 수 있듯이 대부분의 경우 적은 하드웨어의 추가로 최대 고장 검출률을 얻을 수 있었다. 여기서 의사 무작위 테스트는 앞에서 제안된 LFSR/SR 구조를 사용하였으며, 최대 100만개의 패턴을 가졌는데 최대의 고장 검출률을 얻을 수 있을 때까지 고장 시뮬레이션을 수행하였다. 표에서 알 수 있듯이 의사 무작위 테스트로 최대의 고장 검출률을 얻는 것은

표 4 의사 무작위 기법과 혼합 테스트 기법의 비교
Table 4 The comparison of pseudo-random test and mixed test

		의사 무작위 테스트		혼합테스트		% 테스트 수 비교 (혼합/무작위)
		테스트 길이	고장 검출률	테스트 길이	고장 검출률	
조 합 회 로	c432	608	100	262	100	43.09
	c499	1056	100	296	100	28.03
	c880	11456	99.04	754	100	6.58
	c1355	2880	100	2026	100	70.35
	c1908	10304	100	3022	100	29.33
	c2670	60448	91.47	3609	100	5.97
	c3540	27552	99.87	2253	100	8.18
	c5315	5440	100	1128	100	20.74
순 차 회 로	c6288	192	100	101	100	52.60
	s444	544	100	228	100	41.91
	s510	640	100	293	100	45.78
	s526	8160	97.11	2762	100	33.85
	s641	45952	98.27	2538	100	5.52
	s713	45952	98.45	2538	100	5.52
	s820	9920	97.29	4255	100	42.89
	s832	10016	96.96	4653	100	46.46
	s838	54240	90.43	1624	100	2.99
	s953	41216	99.91	5838	100	14.16
	s1196	33600	99.52	5838	100	17.38
	s1238	33600	99.46	6222	100	18.52
	s1423	34752	100	1455	100	4.19
	s1488	4864	100	2053	100	42.21
	s1494	4864	100	2053	100	42.21

상당수의 테스트를 필요로 한다는 것을 알 수 있다. 그리고 경우에 따라서는 s526 등과 같이 최대의 고장 검출률을 얻지 못할 수도 있었다. 이에 반해 혼합 테스트를 사용한 경우는 모든 회로에 대해서 적은 테스트로 최대의 고장 검출률을 얻을 수 있었다. 70.35에서 2.99 %의 적은 패턴수로 100% 고장 검출률을 얻을 수 있었다. 여기서 대부분 절반 이하의 적은 패턴으로 더 높은 고장 검출률을 얻었다. 이것은 짧은 테스트 시간으로 최대 고장 검출률을 확실하게 보장한다는 것을 의미한다. 즉, 무작위 테스트로 검출하기 어려운 무작위 저항 고장을 혼합 테스트 기법으로 모두 잡을 수 있었다는 것을 알 수 있다. 일반적으로 집적화 되는 전자 회로에서 그것의 테스트하는 것이 어려워진다는 것을 고려할 때, 짧은 시간으로 높은 고장 검출률을 얻을 수 있다는 결과는 매우 중요하다.

6. 결 론

본 논문에서 제안된 BIST 혼합 테스트 구조는 테스트 대상 회로에 FSM으로 결정 테스트를 가하고 그 후에 의사 무작위 테스트를 가하는 것이다. 따라서 복잡하고 실질적인 고장 모델에 대해 적은 테스트 패턴 수를 가지고 최대의 고장 검출률을

얻을 수 있다. 그리고 생성기의 면적을 줄이기 위해 몇몇 플립 플롭들은 공유된 구조를 사용하였다. 만일 모든 테스트를 결정 테스트로 수행한다면 테스트 시간이나 테스트 당 검출 고장 수에 있어서 그것이 가장 효율적인 방법이 되겠으나, 실제로 그를 위한 하드웨어 오버헤드가 너무 커서 실용화는 현재로서는 불가능하다고 할 수 있다. 반면에 여기서 제시된 혼합 테스트 기법을 사용한다면 혼합 테스트 생성기의 면적 오버헤드를 줄이면서 최대의 고장 검출률을 보장할 수 있었다. 물론 테스트 길이와 오버헤드 사이에서 적절한 트레이드 오프가 이루어져야 함이 중요한 핵심이다. 전반적으로 큰 회로에 대해 적은 하드웨어의 추가로 높은 고장 검출률을 적은 테스트로 얻을 수 있었다. 예를 들면 c6288은 10.29%의 하드웨어 추가로 총 1028개의 테스트에 의해 100%의 고장 검출률을 얻었다. 만일 더 적은 하드웨어 추가를 필요로 한다면, 좀더 긴 혼합 테스트를 테스트 대상 회로에 가하면 가능하다. 오버헤드가 큰 경우를 해결하기 위해 무작위 부분을 강화시키는 방법을 생각 할 수 있다. 이를 위해 가중 무작위 테스트와 결정 테스트를 동시에 수행하는 혼합 테스트 기법 등을 생각 할 수 있을 것이다.

본 연구는 산학협동재단의 연구비 지원에 의해 수행되었습니다.

참 고 문 헌

- [1] L. T. Wang and E. J. McCluskey, "Circuits for Pseudo-Exhaustive Test Pattern Generation", IEEE Trans, Computer Aided Design, Vol - 7, No 10. 1988.
- [2] P. H. Bardell, W Mcanney and J. Savir, "Built-in Test for VLSI : Pseudo-Random Techniques", John Wiley and Sons, New-York, 1987.
- [3] D. K. Pradhan, "Fault Tolerant Computing Vol. 1-2" Prentice-Hall, 1986.
- [4] Rajagopalan Srinivasan, Sandeep K. Gupta and Melvin A. Breuer, "Novel Test Pattern Generators For Pseudo-Exhaustive Testing", Proc. ITC, Paper 45.4, pp. 1041-1050, 1993.
- [5] Miron Abramovici, "Digital Systems And Testable Design" IEEE Press, 1990.
- [6] M. E. Aboulhamid and E. Cerny, "A class of Test Generators for Built-in Testing", IEEE Trans. on Comp., pp. 977-969, 1983.
- [7] V. K. Agarwal and E. Cerny, "Store and Generate Built-In Testing Approach", Proc. FTCS 11, pp. 35-40, June 1981.
- [8] R. Dandapani, J. Patel and J. Abraham, "Design of Test Pattern Generators for Built-In Test", Proc. ITC, pp. 315-319, 1984
- [9] J. Van Sas, F. Cattthoor and H. De Man, "Cellular Automata Based Self-Test for Programmable Data Paths", Proc. ITC, pp. 769-778, 1990.
- [10] C. Dufaza and F.F.C. Lew Yan Voon, "BIST Generator Based on Asynchronous Counter", 6th Workshop on New Directions for Testing, pp. 229-232, May 21-22, 1992.

- [11] W. Daehn and Mucha, "Hardware Test Pattern Generation for Built-In Testing", Proc. ITC, pp. 110-113, 1981.
- [12] C. Dufaza and G. Cambon, "LFSR Based Deterministic and Pseudo-Random Test Pattern Generator Structures", Proc. ETC, pp. 27-34, 1991.
- [13] W. Starke, "Built-in Test for CMOS Circuits" Proc. ITC, pp. 309-314, 1984.
- [14] S. B. Akers and W. Jansz, "Test Set Embedding in A Built-In Self Test Environment" Proc. ITC, Paper 13.2, pp. 257-263, 1989.
- [15] I. Pomeranz, L. N. Reddy and S. M. Reddy, "COMPACTEST: A Method to Generate Compact Tests Sets for Combinational Circuits", Proc. ITC, pp. 194-203, 1991.

저 자 소 개



김 홍 식 (金 弘 植)

1973년 4월 4일생. 1997년 연세대 전기공학과 졸업. 1997년~현재 동 대학원 전기공학과 석사과정



강 성 호 (姜 成 昊)

1986년 2월 서울대 제어계측공학과 졸업. 1992년 The University of Texas at Austin 졸업(공학). 1989년~1992년 Schlumberger Inc. Research Scientist. 1992년~1992년 The Univ. of Texas at Austin Post Doctoral Fellow. 1992년~1994년 Motorola Inc. Senior Staff Engineer. 1994년~현재 연세대 전기공학과 조교수



이 항 규 (李 抗 圭)

1973년 5월 5일생. 1997년 연세대 전기공학과 졸업. 1997년~현재 동 대학원 전기공학과 석사과정