

論文2002-39SD-7-10

## 패턴 집단 생성 방식을 사용한 내장형 자체 테스트 기법

## (Logic Built-In Self Test Based on Clustered Pattern Generation)

康容碩\*, 金玄焯\*\*, 徐一碩\*\*\*, 姜成昊\*\*

(Yong-Suk Kang, Hyun-Don Kim, Il-Suk Seo, and Sungho Kang)

## 요약

본 논문에서는 패턴 집단 생성 방식을 사용한 새로운 내장형 자체 테스트를 위한 테스트 패턴 생성기를 제안하였다. 제안된 기술은 클럭당 테스트 환경에서 작은 하드웨어 크기를 가지면서 미리 계산된 결정 테스트 집합을 가진다. 테스트를 제어하기 위한 회로는 간단하여 자동적으로 합성된다. 새로운 패턴 생성기를 기존의 방법들과 비교한 결과를 ISCAS 벤치마크 회로를 가지고 검증하였다.

## Abstract

A new pattern generator of BIST based on the pattern clustering is developed. The proposed technique embeds a pre-computed deterministic test set with low hardware overhead for test-per-clock environments. The test control logic is simple and can be synthesized automatically. Experimental results for the ISCAS benchmark circuits show that the effectiveness of the new pattern generator compared to the previous methods.

## I. 서론

내장형 자체 테스트 기법(BIST : Built-In Self Test)은 최신의 논리회로를 테스트하기 위한 중요한 방법으로 부각되고 있다.<sup>[1]</sup> BIST를 사용하여 논리회로를 테스트하는 방법은 크게 “클럭당 테스트(test-per-clock)”와

“주사당 테스트(test-per-scan)”로 구분할 수 있다. 클럭당 테스트 방법은 테스트 클럭마다 패턴을 논리회로에 가하는 구조로 일반적으로 추가적인 회로를 필요로 하여 입력 플립플롭과 테스트 대상 회로(CUT : Circuit Under Test) 사이에 지연이 발생하는 단점을 갖는다. 주사당 테스트 방법은 주사-BIST로 표현되기도 하며, LFSR를 통해 생성된 시퀀스를 주사사슬(scan chain)에 직렬적으로 시퀀스한 후 CUT에 테스트 패턴을 가하는 형태를 취한다. 이러한 방법을 사용하며 주사 플립플롭과 CUT 사이에 추가적인 회로를 덧붙이지 않아도 된다. 하지만, 테스트 시간이 아주 긴 단점을 갖는다. 특히, 주사사슬의 길이가 긴 경우에는 치명적인 문제를 갖는다. 적절한 테스트 동작 시간 내에 높은 고장 검출율을 얻기 위해 다양한 연구들이 발표되었다.<sup>[2~6]</sup> 이 연구들이 공통적으로 사용하는 방법은 높은 고장 검출율을 얻기 위해 강고장들(hard faults)을 검출할 수 있는 결정 패턴을 포함시키는 기법이다. 이러한 기법을 혼합 모드 BIST(Mixed-mode BIST)라 부르며 특별한 강고장들을 검출할 수 있는 결정 패턴을 하나 이상의 의사

\* 正會員, LG 電子 電子技術院 시스템 IC 센터  
(System IC Division, SIC R&D Center, LG Electronics Inc.)

\*\* 正會員, 延世大學校 電氣電子工學部  
(Dept. of Electrical Eng., Yonsei Univ.)

\*\*\* 正會員, 三星電子 시스템 LSI 事業部 SOC 研究所  
(SOC Technology TE Group, System LSI Division Samsung Electronics co., LTD.)

※ 이 연구는 2000년도 한국과학재단 연구비 지원에 의한 결과임.(과제번호 : 2000-1-30200-002-3).

接受日字 : 2002年 3月19日, 수정완료일 : 2002年 6月20日

임의 시퀀스에 추가하는 방법을 사용한다.

혼합모드 방식은 제한된 수의 의사임의 패턴들을 사용하여 쉽게 검출 가능한 고장들을 제거하고 소수의 결정 패턴들을 사용하여 남아 있는 강고장 혹은 임의 패턴 저항 고장(random resistant fault)들을 검출한다. 결정 패턴은 칩(chip) 자체에 압축된 형태로 저장한 후 BIST 동작시에 압축을 풀어 사용하던가,<sup>[4,6]</sup> 또는 직접 "비트 고정(bit fixing)"이나 "비트 반전(bit flipping)" 기술을 사용하여 LFSR 시퀀스에 내장하는 방법<sup>[2,3]</sup>이 있다.

혼합모드 BIST에 대한 많은 연구에도 불구하고, 대부분의 기존 기술들은 많은 테스트 동작 시간을 필요로 하던가 LFSR의 크기 및 부가된 회로의 오버헤드 면에서 비효율적이다. 더구나, 거의 모든 이전의 혼합모드 BIST는 목표로 하는 고장 검출율을 얻거나 혹은 결정하기 위해 과도하고 반복적인 ATPG(Automatic Test Pattern Generation) 및 고장 시뮬레이션 과정을 필요로 한다. 마지막으로, 대부분의 기술들은 클럭당 테스트 아키텍처 또는 주사-BIST 중 하나의 방법만을 사용할 수 있으며 두 가지 환경을 모두 지원하지 못한다.

본 논문에서는 패턴 집단화(clustering)에 기반한 새로운 BIST 기술을 제안한다. 결정 패턴들을 몇 개의 집합으로 집단화하고 각 집합의 모든 패턴들은 비슷한 형태를 갖도록 하기 때문에 강고장들을 검출하기 위한 결정 패턴들을 포함하는 집단들(clusters)의 집합의 기본 패턴만을 패턴 생성기를 통해 생성하는 것이 효과적이다. 제안된 방법은 주사를 위해 필요로 하는 하드웨어 외에 부가되는 하드웨어 없이 클럭당 테스트 및 주사당 테스트 환경 모두에 적용 가능하다. 또한 면적 오버헤드 및 테스트 시간이 이전에 발표된 혼합모드 BIST 방법들에 비해 낮다.

## II. 새로운 패턴 생성기

새로운 패턴 생성기는 임의 시퀀스의 많은 패턴이 고장 검출율을 기여하지 못하며 몇몇 비트를 반전함으로써 이미 계산된 결정 패턴으로 치환(mapping)할 수 있다<sup>[3]</sup>는 사실에 기반하여 개발되었다. 수많은 임의 패턴 중에서 한 개의 대상 패턴을 선택하기 때문에 단지 몇 개의 비트만이 다른 값을 갖는 결정 패턴과 임의

패턴의 쌍을 찾을 수 있는 확률이 매우 높다.

기존의 비트반전 방법은 임의 패턴 시퀀스를 생성하기 위해 LFSR을 사용하였으며 대상 테스트 환경은 주사-BIST이다. 새로운 패턴 생성기는 그림 1에서 보는 것과 같이 구현된 임의 패턴 생성기를 기반으로 하여 클럭당 테스트 환경을 위해 개발되었다. 그림 1은 LFSR의 길이(L)가 2 비트이고 입력 주사사슬의 길이(n)가 5 비트인 임의 패턴 생성기의 예를 나타낸 것이다. 표 1은 LFSR 및 입력 주사사슬의 초기 상태가 각각 (01)이고 (00000)인 가정하에 클럭당 테스트 환경에서 생성된 패턴을 나타낸 것이다. 표 1의 첫 번째 칸은 LFSR의 출력으로 생성되는 한 비트의 값을 나타낸 것이다. LFSR만을 사용하여 5 비트의 주사사슬에 가할 패턴을 생성한 경우는 표 1의 두 번째 칸에 나타내었다. 오른쪽 4개의 칸에는 그림 1에 나타낸 패턴 생성기에 의해 생성된 패턴을 나타낸 것이다.

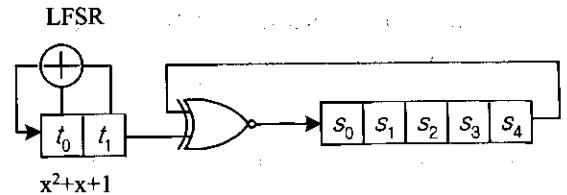


그림 1. 임의 패턴 생성기의 예  
Fig. 1. Example of Random Sequence Generator.

표 1. 그림 1의 생성기에 의한 임의 시퀀스의 예

Table 1. Example of Random Sequence generated by Figure 1.

LFSR 출력	S <sub>0</sub> S <sub>1</sub> S <sub>2</sub> S <sub>3</sub> S <sub>4</sub>				
	LFSR Only	새로운 임의 시퀀스 생성기			
1	10000	00000	10110	11011	00001
0	01000	10000	11011	01101	00000
1	10100	01000	11101	10110	00000
1	11010	00100	11110	01011	.
0	01101	10010	11111	00101	.
1	10110	01001	11111	10010	.
1	11011	10100	11111	01001	.
0	.	11010	01111	00100	.
1	.	01101	10111	00010	.

L과 n의 최대공약수가 1이기 때문에 되먹임(feedback) 경로와 XNOR 게이트가 사용되지 않은 경우 LFSR의 반복 구간의 길이는  $2^L - 1$ 이고 표 1의 두 번째 칸에 나타난 것과 같은 패턴이 생성된다. 두 번째 칸에 굵은 글씨체로 표시된 세 개의 패턴이 계속적으로 반복되어 생성된다. 하지만, 그림 1에 표시된 임의 패턴 생성기의 반복 구간은  $2 \times n \times (2^L - 1)$ 이다. 비록 몇몇 패턴은 중복되어 생성되지만 LFSR만을 사용한 것에 비해 훨씬 다양하고 많은 패턴이 생성됨을 알 수 있다. 그림 1에 표시된 시퀀스 생성기는 일종의 사이클릭 코드 생성기<sup>[7]</sup>로, 이 생성기는 레지스터 기반 설계에 적합한 구조를 갖고 있으며 입력 주사와 출력 주사 레지스터를 분리하여 사용하는 코어에 사용될 수 있다. 물론 테스트 패턴을 가하는 것과 그 응답을 받는 것을 같은 내부의 주사사슬을 사용하는 회로에는 직접 적용하는 것이 불가능하다. 하지만, 이러한 문제를 풀 수 있는 BIST를 위한 효과적인 공간 축소기(space compactor)<sup>[8]</sup> 기술이 여러 개 발표되었다.

그림 2는 결정 패턴을 내장할 수 있는 새로운 패턴 생성기를 나타낸다. 내장 함수(EF : Embedding Function) 회로는 조합 논리회로를 사용하여 구현한다. 주사사슬의 초기화를 위한 아주 작은 회로가 필요하나 용이한 동작 설명을 위해 그림 2에서는 나타내지 않았다. 그림 2에서 CC와 SC는 각각 집단 계수기(cluster counter)와 주사이동 계수기(scan shift counter)를 나타낸다. 결정 패턴 생성을 위한 EF는 대상 패턴을 몇 비트 반전시켜 원하는 결정 패턴으로 치환시키기 위한 큐브들(cubes)의 집합으로 표현된다. 그림 2에서 보는 바와 같이 주사사슬에 패턴을 가하기 위한 두 개의 되먹임 경로가 존재한다. 한 개는 ED를 통한 것이고 다른 하나는 PR를 통한 것이다. 모드 선택 신호 M은 CC의 최하위 비트(LSB : Least Significant Bit)로 EF를 생성하기 위한 상태 변수로 사용되지 않는다. M이 0의 값을 갖는 경우 결정 패턴을 내장시키기 위한 치환 시퀀스가 EF에 의해 주사사슬에 가해진다. 이때 EF는 K 노드에 생성된 패턴의 적당한 비트들을 반전시켜 원하는 결정 패턴이 생성될 수 있도록 한다. 만약, EF의 출력값이 1이면 K의 비트값은 반전되어 ED 노드에 전파된다.

치환 시퀀스가 가해고 한 개의 결정 패턴이 내장된 후, CC의 값이 1 증가하여 M의 값이 1이 되고 SC는

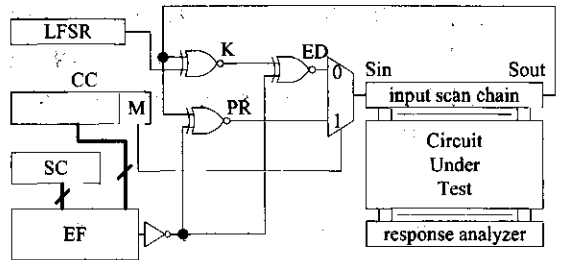


그림 2. 새로운 패턴 생성기  
Fig. 2. New Pattern Generator.

리셋된다. M은 EF를 위한 상태 변수로 사용되지 않기 때문에 EF의 출력 시퀀스는 M이 0이었을 때인 이전 시퀀스와 같다. CC의 최하위 비트인 M의 값이 1인 상태에서는 LFSR에 가해지는 클럭이 차단되고 주사사슬에 저장된 값들이 PR를 통해 회전이동(rotating shift)을 하게 된다. 이전의 M의 값이 0일 때 치환 시퀀스에 의해 반전된 비트들은 다시 반전되어 원래의 패턴으로 복원된다. 결과적으로 M이 1일 때, 복원 시퀀스가 가해진다. 한 개의 집단 시퀀스, 즉, 치환 시퀀스(mapping sequence)와 복원 시퀀스(restoring sequence)가 가해지면 그 후에는 이러한 과정을 반복하게 된다. EF 회로를 생성하기 위해서는 오직 CC와 SC의 상태값만을 사용하였다. LFSR의 상태값은 배선(routing) 오버헤드를 줄이기 위해 사용하지 않았다.

LFSR의 길이가 L이고 주사사슬의 길이가 n인 경우 두 숫자의 최대 공약수가 1이면 반복구간의 길이는  $2 \times n \times (2^L - 1)$ 이고 한 개의 집단이 포함하는 패턴의 수는  $2 \times n$ 이 된다. 따라서  $(2^L - 1)$ 개의 집단들이 생성된다. 작은 LFSR를 사용하여도 다양한 시퀀스를 생성할 수 있기 때문에 하드웨어 오버헤드를 줄일 수 있다. 집단의 개수를  $n_c$ 라고 하고 결정 패턴의 수를  $n_d$ 라 가정하자. 한 개의 집단 시퀀스 당 한 개의 결정 패턴이 내장되므로  $n_c$ 는  $n_d$ 에 비해 크거나 같아야 한다. 하지만 치환 시퀀스와 복원 시퀀스가 많은 수의 약고장들(non-hard faults)을 검출할 수 있다. 따라서 내장하고자 하는 결정 패턴들은 여러 번 순서를 변경하여 고장 시뮬레이션을 수행한 후 결정한다. 고장 시뮬레이션을 수행한 후에 약고장들을 검출할 수 있는 결정 패턴의 수가  $n_h$ 이면 집단의 수는  $n_c \geq n_h$ 로 줄일 수 있다.

본 연구에서는 반복적인 고장 시뮬레이션을 수행하지 않았다. 그 대신 치환 시퀀스와 복원 시퀀스가 결정

패턴 큐브를 포함할 수 있도록 점증적으로 내장 함수 ( $EF$ )를 향상시키는 방법을 사용하였다. 다음 장에선 결정 패턴들을 내장하고  $EF$  회로를 자동으로 합성하는 예를 나타내었다.

### III. 결정 패턴 내장 및 회로 합성 예

표 2는 새로운 패턴 생성기에 의해 생성된 패턴의 예를 나타낸다. LFSR은 그림 2에 표시된 것과 같이 주사사슬의 Sout를 통해 되먹임되는 신호를 제어하는 역할을 수행한다. 예에서 사용된 LFSR의 길이는 2 비트이고 LFSR은 (01, 10, 11)의 상태를 주기적으로 반복한다. 예에서 사용된 주사사슬의 길이는 5 비트이고 주사 플립플롭의 값은 ( $s_4s_3s_2s_1s_0$ )로 표시한다. CC와 SC의 상태값들은 표 2에 표시된 것과 같이 각각 ( $c_1c_0$ ) 및 ( $b_2b_1b_0$ )로 표시한다. CC와 SC의 길이는 LFSR의 길이와 주사사슬의 길이에 따라 결정한다. 주사사슬의 초기값은 (1111)이다. 표 2의 "no  $EF$ "라고 표시된 칸은  $EF$  회로가 존재하지 않는 경우에 생성되는 시퀀스를 나타낸다. 그림 2의  $M$  신호는  $c_0$ 이므로  $c_0$ 의 값이 1인 경우에는 LFSR이 동작하지 않는다. 표에서 보는 바와 같이  $c_0$ 의 값이 0인 경우에는 이전 시퀀스의  $s_0$ 의 값과 새로 가해지는 LFSR의 출력값을 XNOR 한 값을  $s_4$ 에 가하고 나머지 주사사슬의 비트들은 한 비트 이동하여 새로운 시퀀스를 생성한다.  $c_0$ 의 값이 1인 경우에는 LFSR은 동작을 멈추고 매 클럭마다 주사사슬의  $s_0$ 가  $s_4$ 로 가해지고 나머지 비트들은 오른쪽으로 한 비트씩 이동하여 새로운 시퀀스를 생성한다.  $c_0$ 의 값이 0일 때를 치환시퀀스, 1일 때를 복원 시퀀스라 한다.

$n_d$ 가 4이고  $n_h$ 가 2라 각각 가정하자. 강고장들을 검출하는 두 개의 결정 패턴을 내장하기 위해 두 개의 집단, 즉  $n_c=2$ 를 사용한다. 결정 패턴 집합은 (11101, 11010, 0X11X, 0X100)이다. 내장하고자 하는 결정 패턴을 선택하는 순서는 결정 패턴 내의 무상관(don't care) 비트의 개수로 결정한다. 가장 적은 수의 무상관 비트를 갖는 결정 패턴을 우선적으로 내장한다. 이는 많은 수의 무상관 비트를 갖는 결정 패턴의 경우 치환 시퀀스와 복원 시퀀스로 포함될 수 있는 확률이 일반적으로 높기 때문이다. 다시 말해, 많은 수의 비트가 0이나 1의 값으로 고정되어 있는 결정 패턴들은 강고장을 검

출할 수 있는 확률이 높다. 결정 패턴을 내장하기 위한 대상 임의 패턴 집합은 각 집단의 절반의 마지막에 위치한 패턴이다. 즉,  $c_1c_0=(00, 10)$ 인 집단의 마지막 패턴으로 표 2에서 대상 패턴 집합은 {01101, 01001}로 T로 표시하였다.

가장 먼저 내장할 결정 패턴으로 (11101)을 선택하면 대상 패턴 집합 중에 가장 적은 수의 비트가 다른 (01101)을 선택할 수 있다. 두 패턴간에는 오직 한 비트가 다른 값을 갖고 있고 그 비트에 해당하는 상태값은 각각 CC의 경우 (00), SC의 경우 (100)이 된다. CC의 LSB( $M=c_0$ )는  $EF$ 의 상태 변수로 사용되지 않기 때문에 ( $c_1, b_2, b_1, b_0$ ) 형태로 표시하면 on-set은 {0,100}이고 off-set은 {0,000, 0,001, 0,010, 0,011}이 된다. 두 개의 집합을 사용하여 논리 최소화(logic minimization)를 수행하면  $EF1=(-1--)$ 이 되고 이는 on-set의 모든 민텀(minterm)을 포함하고 off-set의 모든 민텀은 포함하지 않는 최소의 논리 함수이다. 표 2의 "by  $EF1$ "으로 표시된 칸에  $EF1$ 를 사용하여 생성된 패턴을 나타내었다. 굵은 글씨체로 표시된 값들은  $EF1$ 에 의해 반전된 값을 표시한다.  $EF1$ 을 생성하기 위해 사용된 상태값들은 고정되고  $EF1$ 에 포함되는 모든 민텀들은 생성된 큐브를 유지하기 위해 계속해서 on-set으로 설정된다. 결정 패턴을 내장하기 위해 사용된 집단 시퀀스는 이후 변화되지 않으며 ( $c_1=0$ )인 한 개의 집단이 결정된다. 이 집단에서 ( $c_1c_0=00$ )인 집단의 반에 해당하는 시퀀스가 치환시퀀스가 되고 ( $c_1c_0=01$ )인 다른 반이 복원 시퀀스에 속하게 된다. 내장된 결정 패턴을 제외한 다른 결정 패턴들과 결정된 집단 시퀀스( $c_1=0$ )에 속하는 패턴들을 비교한다. 결정 패턴 (0X11X)는 밑줄이 쳐진 패턴 (01111)을 포함하므로 내장하고자 하는 결정 패턴 집합에서 (0X11X)는 제외한다. 표 2에서 복원 시퀀스의 마지막 패턴은 항상 원래의 값으로 복원되는데 이는  $EF1$ 에 의해 반전된 비트들이  $c_0$ 가 1일 때 Sout-PR-Sin으로 이어지는 되먹임 경로를 통해 다시 반전되기 때문이다. 이러한 메커니즘은 각 집단의 첫 번째 패턴을 예상할 수 있게 하고 각각의 집단에 결정 패턴을 내장하기 쉽게 한다.

다음으로 내장할 결정 패턴은 (11019)이 되고 이에 대응하는 대상 임의 패턴은 (11001)이 된다. 이는 표 2의 "by  $EF1$ "으로 표시된 칸에 T로 표시하였다. 따라서 on-set은 {1,000, 1,001}이 되고 off-set은 {1,010,

1,011, 1,100)이 된다. 생성된 on-set, off-set 및 EF1을 사용하여 논리 최소화를 수행하면 EF2는 {1,-0-}이 된다. 따라서 두 개의 결정 패턴을 내장하기 위한 EF는 {-1-- , 1,-0-}이 된다. 표 2의 마지막 칸은 EF를 사용하여 두 개의 결정 패턴을 내장하는 시퀀스를 나타낸다. 두 번째 집단( $c_1=1$ )에서 아직까지 내장되거나 포함되지 않은 결정 패턴(0X1X0)은 밑줄로 표시된 {00110, 01110}을 포함하기 때문에 원하는 4 개의 결정 패턴을 모두 생성할 수 있게 된다.

이전 연구<sup>[3]</sup>에서는 EF에 의한 변화가 또 다른 결정 패턴을 내장하기 위해서 다시 변경시켜야 하는 경우 XOR 게이트를 주사시슬의 입력단 바로 앞에 추가하는 방법을 사용하였다. 하지만 본 연구에서는, 결정 패턴을 내장하기 위해 새로운 XOR 게이트를 사용해야 하는

경우에는 LFSR의 길이를 한 비트 늘려 시퀀스의 길이를 크게 하여 보다 큰 대상 집합에서 결정 패턴을 내장할 수 있도록 하는 방법을 사용하였다. 따라서 LFSR의 길이를 한 비트 늘려 지금까지 설명한 과정을 수행하여 새로운 EF를 생성한다.

결론적으로, 표 2에 네모 상자 안에 표시된 패턴은 두 개의 내장된 결정 패턴 (11101, 11010)을 나타내고 밑줄로 표시된 패턴들은 나머지 두 개의 결정 패턴 (0X11X, 0X100)을 대체할 수 있는 패턴을 나타낸다. 각 집단 시퀀스의 마지막 패턴은 내장 메커니즘에서 항상 원래의 값을 유지할 수 있기 때문에 이전의 집단에서 변형된 시퀀스가 다음 집단에 전파가 되지 않는다. 이것은 되먹임 경로를 통해 주사시슬에 패턴이 가해지기 전에 반전된 비트가 복원된다는 것을 의미한다.

표 2. 결정 패턴 내장 시퀀스의 예  
Fig. 2. Sequence Example of Embedding Deterministic Patterns.

state			S <sub>4</sub> S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub>			
$c_1 c_0$	$b_2 b_1 b_0$	LFSR	no EF	by EF1	by EF1 & EF2	
00	000	1	11111	11111	11111	
	001	0	01111	01111	01111	
	010	1	10111	10111	10111	
	011	1	11011	11011	11011	
	100	0	T 01101	11101	11101	
01	000	-	10110	11110	11110	
	001	-	01011	01111	01111	
	010	-	10101	10111	10111	
	011	-	11010	11011	11011	
	100	-	01101	01101	01101	
10	000	1	10110	10110	00110	
	001	1	01011	01011	10011	
	010	0	00101	00101	01001	
	011	1	10010	10010	10100	
	100	1	01001	T 11001	11010	
11	000	-	10100	11100	11101	
	001	-	01010	01110	01110	
	010	-	00101	00111	00111	
	011	-	10010	10011	10011	
	100	-	01001	01001	01001	

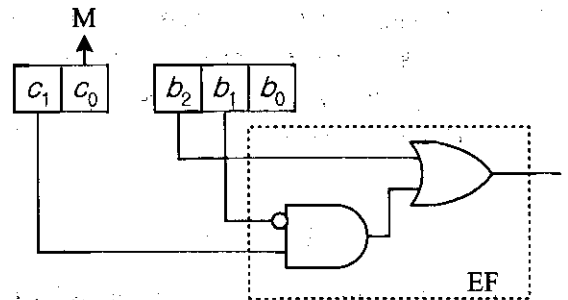


그림 3. EF 회로  
Fig. 3. EF circuit.

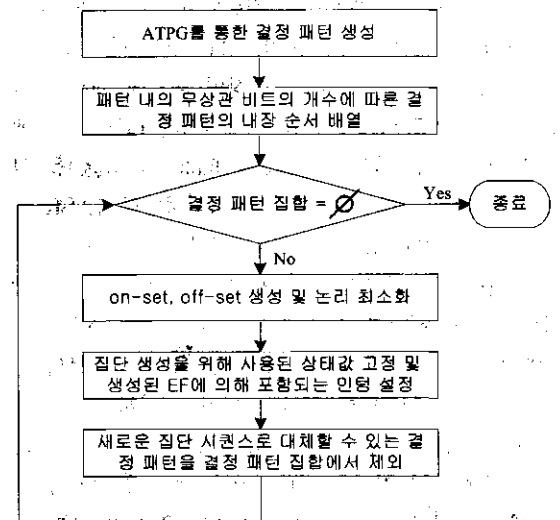


그림 4. EF 회로를 생성하기 위한 알고리즘  
Fig. 4. Algorithm for EC logic generation.

만약 복원 시퀀스를 사용하지 않으면 반전된 값들이 계속해서 주사사슬의 되먹임 경로를 통해 회전하게 된다. 이것은 다음 결정 패턴을 내장하고자 하는 대상 임의 패턴을 찾을 수 있는 확률을 낮추게 된다. 본 장에 나타낸 예에서 알 수 있듯이 반복적인 ATPG 과정이나 고장 시뮬레이션 과정이 필요 없다. 그림 3은 표 2에 나타낸 시퀀스를 생성하기 위한 EF 회로를 나타낸다.

본 연구에서 제안한 패턴 생성기는 내부에 XOR 게이트를 필요로 하지 않고 복잡한 배선을 필요로 하지 않기 때문에 레지스터 기반 설계를 위한 전형적인 BILBO-like 방식에 비해 매우 간단하다. 또한 새로운 패턴 생성기는 레지스터 기반 설계를 위해 개발되었으나 플립플롭이 다음 활성 클럭 전이 전에 주사 모드로 전환할 수 있는 것을 보장하는 칩 내의 공간 축소기를 사용하면 주사-BIST에서도 사용 가능하다. 더구나 집단의 개수는 내장하고자 하는 결정 패턴의 유사성이 클수록 줄일 수 있다. EF 회로를 생성하기 위한 알고리즘을 그림 4에 나타내었다.

IV. 실험 결과

ISCAS85 및 ISCAS89 회로의 조합 회로 버전을 사용한 벤치마크 회로 실험 결과를 표 3에 나타내었다. 새로운 패턴 생성기의 효과를 나타내기 위해 10,000개의 의사임의 패턴을 가하여도 검출되지 않는 무해고장(redundant fault)이 아닌 고착고장을 갖는 회로 중에서 주사사슬의 길이가 50 비트 이상인 회로에 대하여 실험을 하였다. 5 비트에서 14 비트까지의 LFSR 길이에 따라 완전 고장 검출율(complete fault coverage)을 얻을 수 있는 면적을 실험을 통해 확인하였다. LFSR 및 EF 회로는 0.25 μm 공정 라이브러리를 사용하여 면적을 계산하였다. 표 3의 첫 번째 및 두 번째 칸은 각각 대상 회로의 이름과 의사 입력의 개수를 나타낸다. 세 번째 칸은 LFSR의 길이를 나타내었다. EF 회로의 면적은 사용된 LFSR의 길이에 따라 변화하기 때문에 LFSR과 EF 회로를 모두 포함한 면적이 최소가 되는 경우의 LFSR의 길이를 나타내었다. 표 3의 마지막 칸은 새로운 패턴 생성기가 최소의 면적을 갖을 때의 논리곱(product) 개수를 나타내었다. 실험은 클럭당 테스트 환경에서 수행되었으므로 전체 테스트 사이클 수는

테스트 패턴의 수와 일치한다. 테스트 사이클 수는  $2 \times n \times (2^l - 1)$ 과 같은 식을 사용하여 계산할 수 있다. 물론 새로운 패턴 생성기에 의해 생성된 패턴을 사용하여 고장 시뮬레이션을 수행하면 식을 통해 계산한 것에 비해 작은 수의 패턴을 갖을 것이다. 이 식은 단순히 사용 가능한 최대한 패턴 수를 의미한다.

표 3. 실험 결과  
Table 3. Experimental Results.

Circuits	PPI	LFSR	Product
s641	54	5	17
s713	54	5	18
s838	66	5	83
s5378	214	9	25
s9234	247	11	193
s13207	700	10	82
s15850	611	10	188
s38417	1664	12	752
s38584	1464	11	111
c2670	233	5	97
c7552	207	9	73

표 4. 비교 실험 결과  
Table 4. Comparison Results.

Circuits	% of 32 비트 LFSR		리터럴 개수	
	[3]	New	[4]	New
s641	59.7	31.7	52	43
s713	58.2	32.5	46	46
s838	102.0	83.9	209	342
s5378	88.6	44.7	23	75
s9234	510.0	231.7	N/A	994
s13207	179.7	125.9	1955	468
s15850	371.8	236.2	N/A	911
s38417	1698.2	847.9	N/A	4426
s38584	334.1	167.0	157	628
c2670	250.3	100.9	1752	440
c7552	437.0	107.0	2380	301

이전의 결정 패턴 생성 방법<sup>[34]</sup>과의 비교 결과는 표 4에 나타내었다. [3]에서 사용한 방법은 주사-BIST를 목적으로 개발되었기 때문에 아주 큰 테스트 시간을 필요로 한다. 하지만 결정 패턴을 내장하기 위해 조합 논리회로를 사용하였기 때문에 LFSR과 결정 패턴 내장을 위해 사용된 부가 회로의 면적 오버헤드를 본 연구에서 제안한 방법의 면적 오버헤드와 직접 비교할 수 있다. 본 연구와 [3]의 면적은 32 비트 LFSR의 면적을 기준으로 하여 비교하였다. 표 4에서 보는 바와 같이 모든 회로에 대하여 [3]에서 필요로 하는 면적 보다 작은 면적을 사용하여 완전 고장 검출율을 얻을 수 있다. 예를 들어 c7552 회로의 경우 본 연구의 전체 면적 오버헤드는 32 비트 LFSR에 비해 107.0%의 크기를 갖는데 반해 기존 연구<sup>[3]</sup>는 437.0%의 면적을 갖는다. [4]의 연구는 클럭당 테스트 방법을 사용하였으나 완전 고장 검출율을 얻기 위해 초기치 변경(reseeding) 방식을 사용하였기 때문에 본 연구와 직접적인 면적 비교가 불가능하다. 따라서 간접적으로 면적의 크기를 비교하는 방법으로 리터럴(literal)의 개수를 비교하는 것이 가능하다. 일반적으로 논리회로의 면적은 리터럴의 개수에 비례하기 때문에 이를 통한 면적 비교는 실제 회로의 면적 비교와 거의 일치한다. 표 4에서 보는 바와 같이 s838과 s38584를 제외한 모든 회로에 대하여 새로운 패턴 생성 방식이 보다 적은 면적을 필요로 한다. [4]의 연구에서는 큰 벤치마크 회로에 대하여 필요로 하는 패턴의 수를 나타내지 않았기 때문에 테스트에 필요로 하는 사이클 수를 비교하는 것이 불가능하다. 하지만 [4]에서 발표된 작은 회로에 대한 사이클 수를 비교하면 본 연구의 접근 방식이 보다 적은 수의 패턴을 필요로 하는 것을 알 수 있다.

## V. 결 론

본 연구에서는 패턴 집단화를 이용한 새로운 혼합모드 BIST 패턴 생성기를 위한 기법을 발표하였다. 제안된 방법은 특히 레지스터 기반 설계에 적합하게 고안되었으며 내부의 주사 회로에 XOR 게이트나 복잡한 배선을 필요로 하지 않기 때문에 기존의 BILBO-like 접근 방식과 달리 오버헤드가 작다. 또한 경계 주사(boundary scan)를 이용한 설계 방식에도 적용하기 용이하다. 완전 고장 검출율을 얻기 위한 테스트 시간이

짧고 성능 오버헤드와 하드웨어 오버헤드 또한 낮다.

벤치마크 회로들에 대한 새로운 패턴 생성기의 실험 결과에서 새로운 방식이 최근에 발표된 방법들에 비해 BIST 적용에 보다 효과적임을 알 수 있다. 더구나, 반복적인 ATPG 및 고장 시뮬레이션의 수행 없이 결정 패턴 집합만으로 자동으로 내장 회로를 합성할 수 있다.

## 참 고 문 헌

- [1] M. Abramovici, M.A. Breuer and A. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, 1990.
- [2] N. A. Touba, and E. J. McCluskey, "Bit-fixing in pseudorandom sequences for scan BIST", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 20, pp. 545~555, April 2001.
- [3] G. Kiefer and H.-J. Wunderlich, "Using BIST Control for Pattern Generation", *Proc. of IEEE International Test Conference*, pp. 347~355, 1997.
- [4] K. Chakrabarty, B. T. Murray and V. Iyengar, "Deterministic Built-In Self Test Pattern Generation for High-Performance Circuits using Twisted-Ring Counters", *IEEE Trans. on VLSI Systems*, Vol. 8, No. 5, pp. 633~636, October 2000.
- [5] C. Fagot, P. Girard and C. Landrault, "On Using Machine Learning for Logic BIST", *Proc. of IEEE International Test Conference*, pp. 338~346, 1997.
- [6] S. Hellebrand, H.-G. Liang and H.-J. Wunderlich, "A Mixed Mode BIST Scheme Based On Reseeding of Folding Counters", *Proc. of IEEE International Conference*, pp. 778~784, 2000.
- [7] W. W. Peterson and E. J. Weldon, Jr., *Error-correcting codes*, The MIT Press, 1981.
- [8] B. Pouya and N. A. Touba, "Synthesis of zero-aliasing elementary-tree space compactors",

Proc. of IEEE VLSI Test Symposium, pp. 7  
0~77, 1998.

저 자 소 개



康容碩(正會員)

1995년 2월 연세대학교 전기공학과 졸업. 1997년 8월 연세대학교 전기공학과 졸업 (석사). 2002년 2월 연세대학교 전기전자공학과 졸업 (박사). 현재 LG 전자 시스템 IC 사업부 System IC R&D 센터



徐一碩(正會員)

2000년 2월 광운대학교 전자재료공학과 졸업. 2002년 2월 연세대학교 전기전자공학과 졸업 (석사). 현재 삼성전자 시스템 LSI 사업부 SOC 연구소



金玄焯(正會員)

2001년 2월 연세대학교 전기공학과 졸업. 현재 연세대학교 전기전자공학과 석사과정.



姜成昊(正會員)

1986년 2월 서울대 공대 제어계측공학과 졸업. 1988년 5월 The University of Texas at Austin. 전기 및 컴퓨터공학과 졸업(석사). 1992년 5월 The University of Texas at Austin. 전기 및 컴퓨터공학과 졸업(공학). 미국 Schlumberger 연구원, Motorola 선임 연구원. 현재 연대 공과대학 전기전자공학과 부교수