

論文2000-37SD-8-11

# 입력신호 그룹화 방법에 의한 BIST의 테스트 시간 감소

## (Test Time Reduction of BIST by Primary Input Grouping Method)

張允碩\*, 金東郁\*\*

(Yoon-Seok Chang and Dong-Wook Kim)

### 요 약

집적도 증가에 따라 비용이 증가하는 가장 대표적인 분야가 테스트 분야이며, 하드웨어 비용의 상대적인 감소에 따라 BIST 방법이 미래지향적 테스트 방법으로 주목받고 있다. 이 방법이 가지는 가장 큰 단점은 만족할 만한 고장검출률을 얻기 위해 필요한 테스트 시간의 증가이다. 본 논문에서는 BIST의 실현에 있어서 테스트 시간을 감소시키는 방안을 제안하였다. 이 방법은 입력의 그룹화와 테스트 포인트 삽입 방법을 사용하며, 테스트 포인트는 기존에 사용하던 것과는 다른 새로운 정의에 의해 결정된다. 즉, 회로내부 노드 중 입력의 그룹화에 기준이 되며 그룹화된 신호가 병합되는 점들을 기초로 하여 정의된다. 제안한 방법의 주요 알고리즘들은 C-언어로 구현되었으며, 여러 가지 대상회로를 통해 실험한 결과 의사-무작위 패턴을 사용하는 경우에 비해 최대  $10^7$  정도의 테스트 시간 감소를 가져올 수 있었으며, 고장검출률 또한 기존의 BIT방법보다 큰 것으로 확인되었다. 제안한 방법의 대상회로에 대한 상대적인 하드웨어 오버헤드는 대상회로가 커질수록 감소하고 지연시간 증가는 대형회로의 지연시간에 비해 미미한 것이어서, 대형회로를 BIST 방법에 의해 테스트할 때 제안한 방법이 매우 효과적일 것으로 사료된다.

### Abstract

The representative area among the ones whose cost increases as the integration ratio increases is the test area. As the relative cost of hardware decreases, the BIST method has been focused on as the future-oriented test method. The biggest drawback of it is the increasing test time to obtain the acceptable fault coverage. This paper proposed a BIST implementation method to reduce the test time. This method uses an input grouping and test point insertion method, in which the definition of test point is different from the previous one. That is, the test points are defined on the basis of the internal nodes which are the reference points of the input grouping and are merging points of the grouped signals. The main algorithms in the proposed method were implemented with C-language, and various circuits were used to apply the proposed method for experiment. The results showed that the test time could be reduced to at most  $1/2^{40}$  of the pseudo-random pattern case and the fault coverage were also increased compared with the conventional BIST method. The relative hardware overhead of the proposed method to the circuit under test decreases as the size of the circuit to be tested increases, and the delay overhead by the BIST utility is negligible compared to that of the original circuit. That means, the proposed method can be applied efficiently to large VLSI circuits.

\* 學生會員, \*\* 正會員, 光云大學校 電子材料工學科  
(Dept. of Electronic Materials Eng., Kwangwoon Univ.)

接受日字:1999年2月6日, 수정완료일:2000年7月6日

### I. 서 론

반도체 제조공정기술의 발달은 설계에서 TTM(time-to-market)까지의 과정 중 많은 비용의 증가를 초래하

고 있으며, 그 중 가장 대표적이라 할 수 있는 것이 제조된 IC의 테스트 분야이다<sup>[1][2]</sup>. 집적도 증가에 따른 테스트 생성 및 ATE 사용비용의 증가는 미래의 테스트 전략으로 DFT (Design-for-Test) 분야를 지목하고 있으며, 그 중 가장 널리 연구되고 응용되고 있는 분야는 BIST(Built-In Self-Test)<sup>[1]-[4]</sup>이다.

BIST는 테스트를 위한 장비를 IC내부에 내장(built-in)시켜 외부에서의 큰 조작없이 자체적으로 테스트를 수행(self-test)시키는 방법으로, 그림 1에 BIST를 사용하는 IC의 기본형태를 나타내었다. CUT(Circuit Under Test)는 IC가 가져야 할 동작들을 포함하고 있으며, 그 입력에 CUT 테스트를 위한 테스트 입력을 생성하는 테스트 패턴 생성기(Test Pattern Generator, TPG)가 부가되고, 출력단에는 테스트 응답의 비교과정을 최소화하기 위한 테스트 응답 처리기(Test Response Processor TRP)가 내장된다. TRP는 주로 테스트 데이터를 압축하는 기능을 가져 비교대상 데이터의 양을 줄이는 역할을 한다. TPG에서 생성되는 패턴으로는 결정(deterministic) 패턴, 무작위(random) 패턴, 그리고 의사-무작위 패턴(Pseudo-Random Pattern, PRP)이 주로 사용되나, 그 중 PRP가 가장 널리 사용되고 있다. TPG와 TRP를 실현하는 기술은 1970년대 후반부터 많은 연구가 이루어지고 있으나, 최근 가장 널리 사용되는 것은 선형피환 쉬프트 레지스터(Linear Feedback Shift Register)<sup>[5]</sup>와 셀룰러 오토마타(cellular automata)<sup>[6]</sup>이다. 그 중 셀룰러 오토마타는 현재 연구가 진행 중이며 현재까지 가장 많이 사용되고 응용되고 있는 것은 LFSR이다.

BIST는 자체 테스트 기능을 포함하므로 ATE의 사용이나 테스트 응답에 의한 고장여부 판단 등에 있어서 매우 많은 비용을 절감할 수 있으며, 실제 회로가 동작하는 속도로 테스트를 수행할 수 있는 장점을 갖고 있어 많은 부분에 응용되고 있다. 최근에는 조합논리 뿐 아니라 순차적인 동작을 위한 BIST<sup>[7]</sup>, 지연고장을 위한 BIST<sup>[8]</sup>, 단락고장을 위한 BIST<sup>[9]</sup> 등 많은 응용분야에서 적용되고 있다. 그러나 BIST를 적용하는 데는 크게 두 가지의 단점이 지적되고 있다. 그 첫 번째는 부가 하드웨어의 양이다. 입력측과 출력측에 테스트를 위한 회로를 부가함으로써 칩의 면적을 원래의 동작 뿐 아니라 테스트를 위한 동작에도 할애를 하여야 하며, 입·출력단의 부가회로로 인한 칩의 성능저하를 가져올 수 있다는 것이다. 이 문제는 하드웨어의 상

대적인 비용감소로 하드웨어 부가에 대한 상당부분이 받아들여지고 있으며, 신호경로의 우회통과 및 동작속도의 증가로 성능저하 문제도 많은 해결방안이 발표된 바 있다.

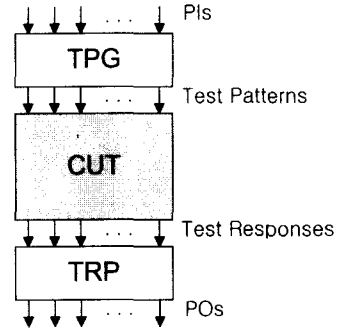


그림 1. BIST를 포함한 IC의 기본구조  
Fig. 1. Fundamental structure of IC with BIST.

또 하나의 문제는 테스트 시간의 증가이다. PRP를 사용하는 경우 거의 고갈패턴과 차이가 없고 입력수가 증가하고 있는 추세에서 보면 비현실적인 테스트 시간이 요구된다(예로써, 50개의 입력을 가진 IC의 경우 100MHz의 테스트 클럭을 사용한다 하여도 약 130일의 테스트 시간이 소요됨). 이 문제를 해결하는 방법으로 회로를 분할하는 방법<sup>[10]</sup> 등이 발표된 바 있으나, 이 또한 분할된 회로의 연결선 등에 따른 부가 하드웨어의 증가 및 테스트 시간의 감소정도가 아직 비현실적이어서 큰 효과를 거두지 못하고 있다<sup>[11]</sup>.

본 논문에서는 테스트 시간의 감소에 초점을 맞춘 BIST 구현방법을 제시하고자 한다. 이 방법은 입력신호들을 종속성에 따라 구분하여 그룹화하는 방법을 사용하며, 회로 내부노드들의 일부분을 테스트를 위한 단자로 사용하여 의사-무작위 패턴 생성방법을 사용하면 테스트 시간을 상당히 감소시키고 고장검출률을 향상시키는 결과를 초래한다. 본 논문에서는 BIST의 도구로서 LFSR을 사용하며, 대상 회로 레벨은 비규칙적인(random) 게이트 레벨 회로로 한다. 본 논문에서 제시된 방법을 실현한 후 여러 가지의 회로를 대상으로 제시한 방법이 상당한 테스트 시간을 감소시킬 수 있음을 보인다.

## II. LFSR을 이용한 BIST의 특성

본 장에서는 LFSR를 사용하는 BIST 방법의 특성 중 본 논문과 관련된 사항을 피력한다.

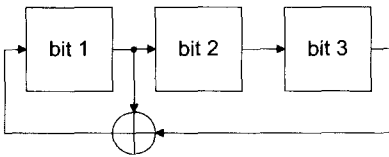
1. LFSR에 의한 의사-무작위 패턴

비규칙적인 논리회로의 BIST 실현에 있어 일반적으로 가장 널리 사용되는 입력패턴은 의사-무작위 패턴이며, 이를 구현하는 방법에 있어서도 LFSR를 가장 많이 사용하고 있다. 본 논문에서도 LFSR를 사용하는 TPG와 TRP를 사용하며, 그 패턴의 종류도 PRP를 기본적으로 사용한다.

**[정리 1]** n 비트 LFSR의 특성다항식을 primitive 다항식으로 구현하고 LFSR의 초기값으로 모든 비트가 0이 아닌 값을 사용하면 모든 비트가 0인 패턴을 제외한  $2^n-1$ 개의 서로 다른 n 비트 패턴(의사-무작위 패턴)이 반복적으로 생성된다<sup>[12]</sup>.

이 정리는 매우 잘 알려져 있으므로, 증명은 생략한다.

**[정리 2]**  $2^n-1$ 개의 n 비트 의사-무작위패턴 세트는 n 비트 중 임의로 선택한  $k(k \leq n)$  비트의 의사-무작위패턴을 포함한다.



(a)

clock	bit 1	bit 2	bit 3
scan	0	0	0
0(seed)	1	0	0
1	1	1	0
2	1	1	1
3	0	1	1
4	1	0	1
5	0	1	0
6	0	0	1
7	1	0	0

(b)

그림 2. Primitive 다항식을 가진 3 비트 LFSR의 패턴 생성

- (a) 3-비트 LFSR,
- (b) (000)를 포함한 출력패턴

Fig. 2. Pattern generation of 3 bit LFSR with primitive polynomial.

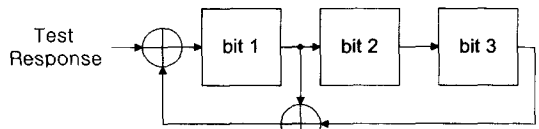
- (a) 3-bit LFSR,
- (b) Output patterns including (000)

이 정리에 대한 증명은 [13]을 참조하기 바람이며 여기서는 역시 생략한다. 그림 2의 예제에서 보면 3 비트 중 어느 2개를 선택하여도 7개의 패턴에는  $2^2-1=3$ 개의 2 비트 패턴이 모두 포함되어 있음을 쉽게 알 수 있다.

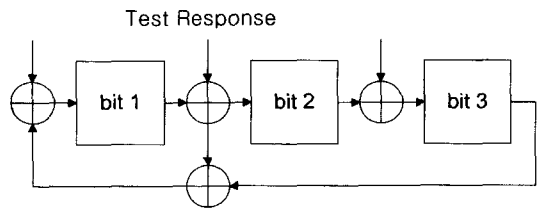
2. LFSR의 고장검출률

테스트 응답의 압축을 위해 LFSR를 사용할 때[14] 단일입력(Single-Input Signature Register)의 경우와 다중입력(Multiple-Input Signature Register, MISR)의 경우로 구분되며, 그림 3에 두 형태를 3 비트 출력에 대해 나타내었다. SISR의 경우 다출력 회로의 출력을 압축하기 위해서는 출력을 다중화기(Multiplexer, MUX)를 통해 단일비트 출력으로 변환하여야 하므로 테스트 시간이 상대적으로 길어지고 실시간 테스트가 불가능할 수 있다. 따라서 일반적으로는 그림 3 (b)의 MISR을 사용한다. 이 경우 출력수 만큼의 LFSR 비트수가 필요하며 각 단의 입력부에 각 출력의 응답을 XOR를 통해 입력한다.

MISR은 m(의사-무작위 패턴의 경우  $m=2^n-1$ ) 개의 n 비트 입력패턴에 대해 k 출력 응답(총  $n \times m \times k$  비트)을 k 비트로 압축한다. 따라서 고장에 대한 정보를 소실(alias)할 수 있으며, 이로 인해 고장검출률에 문제가 발생할 수 있다. MISR에 대한 고장검출률의 계산 및 고장검출률을 개선하기 위한 많은 연구가 진행되어 왔다<sup>[14]</sup>. 일반적으로는 다음의 정리 3에 의한 고장검출률이 통용되고 있다.



(a)



(b)

그림 3. LFSR를 사용한 테스트 응답 압축기

- (a) SISR, (b) MISR

Fig. 3. Test response compactor using LFSR.

- (a) SISR, (b) MISR

[정리 3] k 비트 LFSR이 MISR의 기능을 담당할 경우 n 개의 입력패턴이 인가되었을 때 고장정보를 소실할 확률은 다음과 같다.

$$P(\text{Alias}) \approx \frac{2^{n-k}-1}{2^n-1} \quad (1)$$

이 정리의 증명 역시 기존의 많은 논문에서 찾을 수 있으므로 본 논문에서는 생략한다. 식 (1)에서 일반적으로는 출력수 k보다 입력패턴수 n이 매우 크므로 다음과 같이 근사할 수 있다.

$$P(\text{Alias}) \approx \frac{1}{2^k} \quad (2)$$

식 (1) 또는 (2)에 의하면 고장검출률은 MISR의 비트수를 증가할수록 증가함을 알 수 있다.

### III. 입력의 그룹화

이 장에서는 본 논문에서 테스트 시간을 줄이기 위해 제안되는 입력의 그룹화 방법에 대해 설명한다. 먼저 본 논문에서 사용하는 용어들을 정의하며, 좀 더 구체적인 설명을 위해 그림 4의 간단한 논리회로를 예로 들어 설명한다.

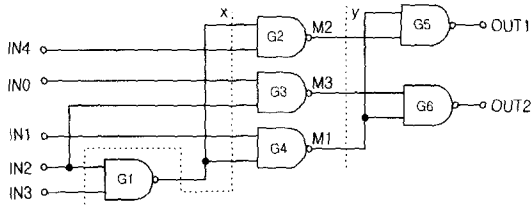


그림 4. 예제 논리회로  
Fig. 4. An example logic circuit.

<정의 1> 논리회로에서 주어진 PI(IN<sub>j</sub>)에 대한 한 내부노드(N<sub>i</sub>)의 깊이(Depth<sub>IN<sub>j</sub></sub>(N<sub>i</sub>))는 IN<sub>j</sub>에서 N<sub>i</sub>까지 통과하여야 하는 게이트 수로 정의한다. 또한 N<sub>i</sub>의 최소 깊이(MinDepth(N<sub>i</sub>))는 각 PI에서 N<sub>i</sub>까지의 깊이 중 최소값으로 정의한다. 즉,

$$Depth_{IN_j}(N_i) = \text{Number of gates between } IN_j \text{ and } N_i \quad (3)$$

$$MinDepth(N_i) = \min\{Depth(N_i) \text{ for any primary input}\} \quad (4)$$

그림 4에서  $Depth_{IN4}(M2)=1$ 이며,  $Depth_{IN2}(M1)=2$

다. 그러나  $MinDepth(M1)=1$ 이다.

<정의 2> 논리회로에서 기준점(reference point, RP<sub>i</sub>)을 회로의 내부노드(i) 중 하나로 정의한다. 또한 최소 깊이 k인 기준점들의 집합을 깊이 k인 기준선(reference line with MinDepth=k, RL(k))이라 하며, 기준선은 회로를 완전히 이등분한다. 즉,

$$RL(k) = \{all RP_i \mid RP_i = \text{reference point, } MinDepth(RP_i) = k\} \quad (5)$$

그림 4의 예제에서 M1과 M2는 기준점이 될 수 있으며, 선 y는 RL(1)의 기준선을 나타낸 것이다.

일반적으로 비규칙적인 논리회로에서는 모든 주입력(Primary Inputs, PIs)이 모든 주출력(Primary Outputs, POs)에 영향을 미치지 않는 경우가 대부분이다. 더욱이 입력에서 출력으로의 신호경로 중간의 기준점에 대해서는 주입력 중 일부분만 그 노드의 논리값에 영향을 미치는 경우가 많다. 예를 들어 그림 4에서 OUT1은 {IN1, IN2, IN3, IN4}에만 종속되고 OUT2는 {IN0, IN1, IN2, IN3}에만 종속된다. 또한 M1은 {IN1, IN2, IN3}에만 영향을 받고 M2는 {IN2, IN3, IN4}에만 종속된다. 만약 주어진 기준선이 정해지면 그 기준선 내의 각 기준점에 영향을 미치는 PI들이 정해진다.

<정의 3> 주어진 기준선(RL(k))에 대해 기준선 내의 각 기준점(RP<sub>j</sub>)에 영향을 미치는 PI들의 집합을 그 기준점에 대한 입력그룹(IG(RP<sub>j</sub>))이라 정의한다. 또한 기준선(RL(k))에 대한 입력그룹(IG(RL(k)))은 기준선 내의 모든 기준점들에 대한 입력그룹의 합집합으로 정의하며, 여기에는 모든 PI를 포함한다.

그림 4에서 기준선 y 내에는 M1, M2, M3가 포함되며, IG(M1)={IN1, IN2, IN3}, IG(M2)={IN2, IN3, IN4}, IG(M3)={IN0, IN2}로 각각 나타낼 수 있으며, IG(y)={IN0, IN1, IN2, IN3, IN4}로 모든 PI들을 포함한다.

본 논문에서는 이 기준선에 의한 각 기준점들의 입력그룹화를 기준으로 입력들을 분할하여 BIST를 실현함으로써 테스트 시간을 감소시키고자 한다. 본 논문에서는 MinDepth=1인 기준선을 선택하며 이 기준선의 조건이 외에 한 가지의 조건을 추가하여 입력을 그룹

화한다. MinDepth=1인 기준선내의 특정 기준점은 모든 PI에서의 깊이가 1이라고 할 수는 없다. 즉 그 기준점 Mi에서의 최소 깊이는 1이지만 DepthPIj(Mi)>1인 PIj가 존재할 수 있다. 이 경우는 PIj와 Mi사이에 다른 게이트들이 존재하는 경우이다. 그림 4의 예제회로에서 DepthIN1(M1)=1이나 DepthIN2(M1)과 DepthIN3(M1)가 모두 2이다. 이 경우는 G4의 입력이 IN1과 G1의 출력이기 때문이다. 본 논문에서는 이러한 경우 Depth>1인 경로를 다시 분석하여 경로상의 게이트에 대해 다시 그룹화한다. 이와 같은 기준에 의해 그림 4의 예제회로를 그룹화한 최종 결과는 식 (6) 같으며, 입력 그룹화 알고리즘은 그림 5에 나타내었다.

$$\begin{aligned}
 IG(G1) &= \{ IN2, IN3 \} \\
 IG(G2) &= \{ IN2, IN3, IN4 \} \\
 IG(G3) &= \{ IN0, IN2 \} \\
 IG(G4) &= \{ IN1, IN2, IN3 \}
 \end{aligned}
 \tag{6}$$

```

InputGroup()
input: logic gate circuit, all PIs
output: PI groups
begin
  find all gates(Gi) with MinDepth(Gi)=1;
  number all Gis (0 ≤ i ≤ n-1);
  for i=0 to n-1
    loop 1
      find all PIs affecting Gi and include into IG(Gi);
      calculate DepthPIj(Gi) for PIs in IG(Gi);
      if any PIj with 0 < DepthPIj(Gi) < ∞ in IG(Gi), then
        loop 2
          find the intermittent gate(Gk);
          find all PIs affecting Gk and include into IG(Gk);
          calculate DepthPIk(Gk) for PIs in IG(Gk);
          if all DepthPIk(Gk)=1 then
            exit loop 2
          else
            goto loop 2
          end if
        end loop 2
      end if
    end loop 1
  end
end
    
```

그림 5. 입력 그룹화 알고리즘  
Fig. 5. Input grouping algorithm.

그림 5의 알고리즘에 의해 PI들이 그룹화된 경우, 기준선에 의한 테스트를 가정한다면 모든 PI들을 대상으로 의사 무작위 패턴을 생성할 필요없이 각 그룹에 해당하는 입력들에 대해 따로 테스트 패턴을 생성하여 사용할 수 있다. 이 때 분류된 각 입력그룹들은 기준선에 대해 서로 독립적이므로 동일한 패턴을 두 그룹에 인가하여도 그 독립성은 그대로 유지된다. 본 논문에서는 이와 같은 성질을 입력패턴의 수를 줄이는 방법으로 사용한다. 즉 분류된 입력그룹중 최고의 크기를 가

진 그룹을 기준으로 의사 무작위 패턴을 생성하고 이 패턴들을 나머지 그룹들이 공유하도록 한다. 이 때 최고의 크기를 갖는 그룹이 하나 이상인 경우는 그 중 임의의 것을 선택하도록 한다. 최고의 크기보다 작은 크기(k)의 그룹들은 최고크기의 그룹에 대해 생성되는 비트 중 임의로 k 비트를 선택한다. 이 경우 정리 2에 의해 k 비트는 고갈패턴을 갖는다. 그림 4의 예제회로에서는 IG(G2)와 IG(G4)가 가장 많은 3개의 요소를 갖고 있으며, 이 두 그룹의 요소는 동일하므로 어떤 것을 기준으로 하여도 결과는 동일하다.

IG(G1)과 IG(G3)는 두 개의 요소를 가지므로 G2 또는 G4의 세 비트 중 임의의 두 비트를 사용한다. 그림 4에 대한 패턴할당의 예를 그림 6에 나타내었다.

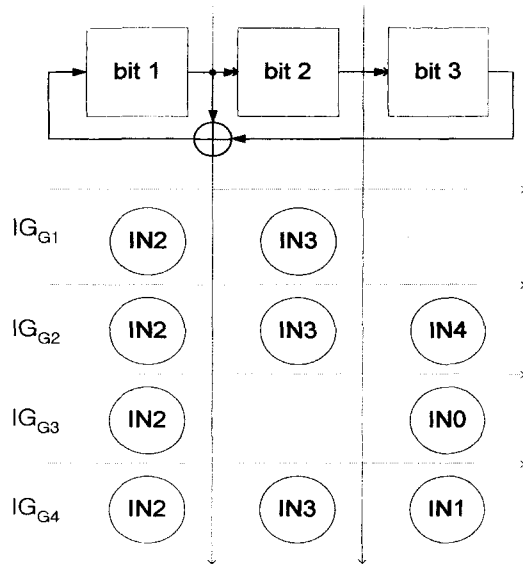


그림 6. 그림 4의 회로에 대한 패턴할당  
Fig. 6. Pattern assignment for the circuit in Fig. 4.

#### IV. 테스트 포인트 삽입

테스트를 위시한 많은 분야에서 기존의 많은 연구는 테스트 포인트를 삽입하는 방법을 사용하여 왔다. 그 중 테스트 분야에 있어서 대부분은 특정 대상회로의 특수성을 감안한 삽입방법[9]이나 상위-레벨의 설계를 합성할 때 특별한 기능을 특정 내부노드에 추가하고자 할 때<sup>[15]</sup> 사용한다. 이 경우 대부분 내부노드들의 테스트 용이성(testability)<sup>[16]</sup>을 계산하여 테스트 포인트들을 찾는다. 즉, 기존의 테스트 포인트들은 제어성/관찰성이

가장 낮은 내부노드들로 정의한다. 그러나, 본 논문에서는 이와는 상이한 정의를 사용한다. 본 논문의 테스트 포인트를 정의하기 위해 다음 용어를 먼저 정의한다.

**<정의 4>** 입력 그룹화에 의해 입력들이 분류된 경우 각 그룹의 신호가 융합하는 점을 *융합점(merging point, MP)*이라 정의한다. 또한 두 개의 동일한 입력 그룹들이 융합하는 점을 *동종 융합점(homogeneous merging point, HMP)*이라 정의한다.

그림 4의 예제에서 OUT1과 OUT2는 모두 MP이며, OUT1은 G1, G2, G4의 MP이고 OUT2는 G1, G3, G4의 MP이다.  $IG(M1)=(IN1, IN2, IN3)=IG(M2)$ 이므로 OUT1은 HMP이다.

**[정리 4]**  $MinDepth=1$ 을 기준으로 입력을 그룹화하고 한 개의 LFSR에 의해 패턴을 생성하여 이를 배분하여 각 그룹의 패턴으로 사용할 경우 HMP를 제외한 MP를 출력으로 하는 게이트의 입력에는 발생가능한 모든 패턴이 입력된다.

(증명) 이 정리의 증명은 자명한 것이어서 간단히 설명으로 대신한다. 기준선의 측면에서 보았을 때 각 입력 그룹은 독립적이며, 비록 단일 LFSR에서 패턴이 생성되어 사용한다 할 지라도 각 그룹의 가능한 모든 입력이 입력되는 것이므로 HMP를 제외한 MP출력으로 하는 입력에는 그 회로 특성상 가능한 모든 발생가능한 입력패턴이 입력된다. 단, HMP를 출력으로 하는 게이트의 입력에는 동일한 패턴이 동일한 순서로 입력되는 경우이므로 각 입력에 동일한 패턴이 발생할 수 있다.

이 정리에 의해 HMP가 아닌 MP 게이트의 입력에는 그 회로에서 발생가능한 모든 벡터가 PI그룹의 의사-무작위 패턴으로 생성되므로 이 MP들은 이 그룹화에 의해 입력패턴에 의한 고장검출률의 손실은 없다. 그러나 HMP들은 회로의 특성을 감안한 발생가능한 패턴의 부분집합만 생성될 수 있으므로 HMP에 대해서는 새로운 입력패턴을 생성하여 입력하는 것이 고장검출률의 손실을 막는 길이다. 본 논문에서는 HMP 게이트의 입력을 테스트 포인트 집합에 포함시켜 각 HMP게이트의 입력에 외부에서 테스트 패턴을 인가하도록 한다.

HMP게이트의 입력에 새로운 테스트 패턴을 인가하기 위해 본 논문에서 사용하는 방법은 입력그룹에 대

해 LFSR로 생성된 패턴을 테스트 포인트에도 인가하는 방법이다. 이 경우 HMP의 개수와 각 HMP 게이트의 입력수를 고려하여야 하는데, 본 논문에서는 다음과 같은 방법으로 테스트 포인트 수 및 테스트 인가방법을 결정한다.

- (1) HMP수가 한 개 이상일 경우, 각 HMP는 서로 독립적이므로 동일한 입력패턴을 각 HMP 게이트의 입력에 인가하여도 고장검출률에는 영향을 미치지 않는다. 따라서 본 논문에서는 각 HMP 게이트의 입력에 입력그룹에 대해 생성된 패턴들을 공통으로 인가한다.
- (2) 한 HMP 게이트의 입력수(k)가 최고크기의 입력그룹에 속한 PI수(n)보다 작은 경우, 입력그룹에 대해 생성되는 n 비트에서 임의로 k 비트를 HMP 게이트의 입력에 인가한다.
- (3) 한 HMP 게이트의 입력수(k)가 최고크기의 입력그룹에 속한 PI수(n)보다 큰 경우, 이 때는 LFSR에서 생성되는 비트수가 작으므로, 본 논문에서는 먼저 k 비트 중 n개를 임의로 할당하고 (k-n)개의 입력에는 n개 중 임의로 (n-k)개를 중복 할당하는 방법을 취한다. 이 방법으로는 해당 HMP 게이트의 입력에 모든 패턴이 입력되지 않아 고장검출률이 저하될 수 있다. 그러나, 이 문제는 부가 하드웨어의 양 및 테스트 시간과 고장검출률이 상호보완적(trade-off)이고 본 논문의 목적이 테스트 시간의 감소이므로 위와 같은 방법을 선택하였다. 참고로 다음 장에 소개될 대사회로들에서는 이와 같은 문제가 발생되지 않았으며, 그 외의 회로에서도 이 문제가 발생할 확률은 매우 적은 것으로 시뮬레이션 결과 나타났다.
- (4) 각 HMP 게이트의 입력수 중 가장 큰 것이 최고크기의 입력그룹에 속한 PI수보다 작을 경우, LFSR에서 생성되는 비트수를 모두 HMP 게이트에 사용할 수 없으므로, 이 때는 HMP가 아닌 MP게이트의 입력에 그 나머지 비트들을 사용할 수 있으며, MP게이트 입력의 테스트 포인트로의 선택은 다음의 우선순위에 의해 결정된다.

(i) 그룹화된 PI수가 많은 MP게이트 입력

(ii) 팬-아웃수가 많은 MP게이트 입력

입력그룹에서 PI수가 많은 MP게이트 입력은 다른 입력보다 더 많은 입력패턴의 변화가 필요하며 팬-

아웃수가 많은 노드는 그 만큼 제어성/관찰성이 나쁘므로 본 논문에서는 이 두 기준에 우선순위를 두었다.

이상에서 설명한 테스트 포인트 결정 알고리즘은 그림 7에 나타내었으며, 이 알고리즘의 결과인 테스트 포인트들에 대해서는 다음 장에서 설명하는 하드웨어를 부과하여 BIST의 도구로 사용한다. 그림 7에 의하면, HMP가 아닌 MP를 테스트 포인트에 포함하는 경우는 테스트 포인트의 수가 충분치 않을 경우에 한하며, HMP가 아닌 MP를 테스트 포인트에 포함시킬 것인지의 여부는 선택적이다. 충분한 수의 테스트 포인트는 설계자의 의도에 따라 결정할 수 있으며, 부과 하드웨어 양과 고장검출률의 상호보완적 관계에 따라 결정된다.

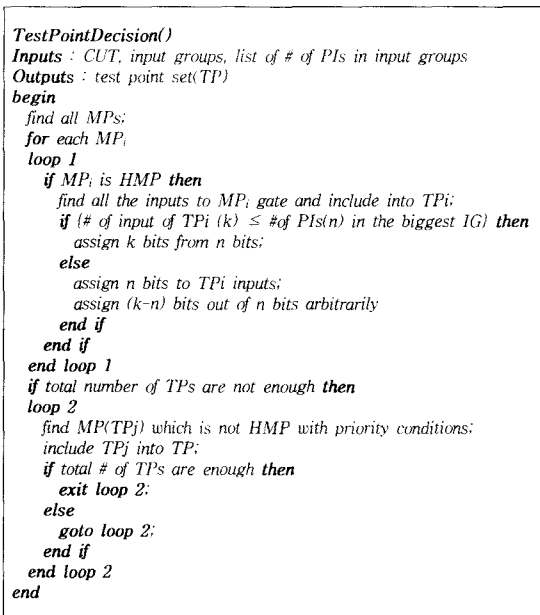


그림 7. 테스트 포인트 결정 알고리즘  
Fig. 7. Determination algorithm of test points.

## V. BIST 실현

III장 및 IV장에서 설명한 알고리즘에 의해 주어진 회로의 BIST 기능을 부과한다. 먼저 PI측의 TPG는 최고크기의 입력그룹에 속한 PI수와 동일한 비트수를 갖는 LFSR로 형성한다. 본 논문에서는 II장에서 설명한 바와 같이 PRP에 모든 비트가 '0'인 패턴을 포함한 고

갈패턴을 사용하므로, TPG는 LFSR 기능 이외에 주사 기능을 포함하여야 한다. 그림 8에 이 TPG를 3 비트의 예로 실현하였는데, 비트 1의 플립-플롭 앞단에 한 개의 MUX를 부가하여 실현한다. 이 MUX는 SCANSEL 신호에 의해 제어되며, SCANSEL=0일 때는 주사기능을 수행하고 SCANSEL=1일 때는 LFSR 기능을 수행한다. 또한 MODE신호는 정상동작과 테스트 동작을 구분하는 신호이며, MODE=0일 때 정상동작을, MODE=1일 때는 테스트동작을 각각 수행한다. 따라서 정상동작 시 본 논문에서 제안한 TPG로 인해 부가되는 지연시간은 한 개의 MUX를 통과하는 시간임을 알 수 있다.

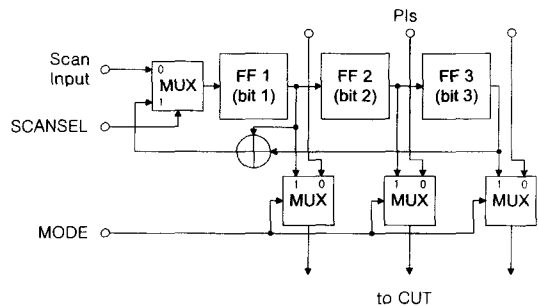


그림 8. 3-비트 TPG  
Fig. 8. 3-bit TPG.

IV장에서 정의한 테스트 포인트에서는 그 전 단의 출력을 받음과 동시에 그 뒤의 회로에 대한 테스트 패턴을 인가하여야 한다. 본 논문에서는 II장에서 언급한 MISR을 테스트 데이터 압축기로 사용한다. 그러나 IV장에서 설명한 대로 테스트 포인트에 제공하여야 하는 테스트 패턴들은 테스트 포인트에서 생성되는 것이 아니고 입력단 TPG에서 생성된 패턴을 사용하므로 테스트 포인트에서의 테스트 생성을 위한 회로는 필요치 않다. 따라서 각 테스트 포인트에는 MISR을 구성할 한 개의 플립-플롭과 정상동작과 테스트 동작을 구분할 한 개의 MUX가 필요하다. 그 외에 테스트 결과를 주사출력하기 위한 주사경로의 설정에 필요한 MUX가 MISR의 첫 번째 비트 전단에 필요하다.

그림 1에 나타낸 기본적인 BIST에서는 MISR을 PO에 부착한다. 본 논문에서는 테스트포인트들의 플립-플롭들로 구성된 MISR과 PO 단에 부착되는 MISR을 통합하여 한 개의 MISR로 구성한다. 이 때 각 PO에는 MISR을 구성하는 한 개의 플립-플롭이 부착된다. 그림 4의 예제회로에 그림 8의 TPG와 테스트 포인트 및

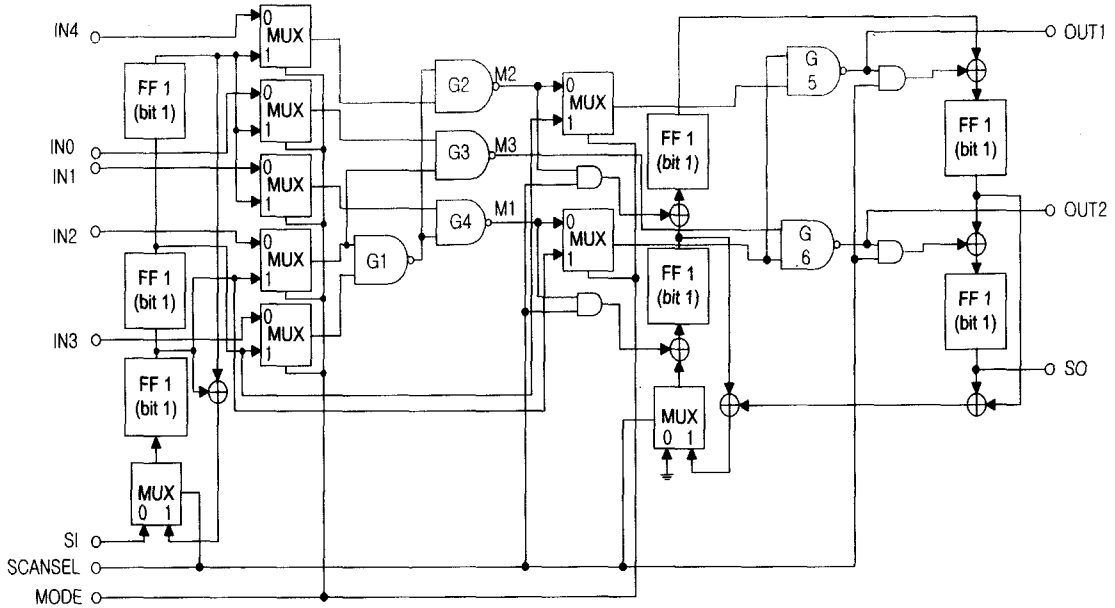


그림 9. 그림 4의 회로의 제안한 BIST 실현 결과  
 Fig. 9. Implemented Result of proposed BIST method to circuit in Fig. 4.

PO의 플립-플롭들로 구성된 MISR을 모두 부착한 회로를 그림 9에 나타내었다. 이 그림에서는 BIST 실현을 위해 네 개의 추가 입력단자를 필요로 하는 것으로 나타내었는데, 이 외에도 테스트 수행에 필요한 테스트 클럭과 압축된 테스트 데이터를 출력하기 위한 주사출력이 추가로 필요할 수 있다(그림에서 각 플립-플롭의 동기신호로 테스트 클럭이 입력되어야 하나, 이를 생략하였음). 추가되는 입·출력 단자들은 그 기능과 함께 표 1에 나타내었다. 이 표에서 TCK는 CUT가 클럭을 사용하는 경우 그 클럭을 TCK로 사용할 수 있고, SI와 SO는 PI중 한 개와 PO 중 한 개를 각각 사용할 수 있으므로, 결국 추가되어야 하는 I/O 핀수는 두 개로 볼 수 있다. 입력측에 세 비트의 LFSR이 PI 그룹과 테스트 포인트들의 테스트 패턴을 생성하고 두 개의 테스트 포인트와 두 개의 PO에 대한 플립-플롭들이 하나의 MISR을 형성하고 있다. 테스트 포인트와 PO를 묶어 하나의 MISR로 형성하는 것은 테스트 포인트의 제어성/관찰성을 증가시키는 것 뿐 아니라 MISR 비트수의 증가에 따르는 고장검출률의 증가효과도 수반하게 된다. 본 논문에서 제안한 BIST의 실현을 위해 정상동작은 두 개의 MUX와 두 개의 AND 게이트의 지연시간의 추가되는 것을 알 수 있다.

표 1. 제안한 BIST를 위해 추가되는 입·출력  
 Table 1. Additional I/Os for the proposed BIST.

Name	Function
TCK	test clock
MODE	normal/test mode selection
SCANSEL	scan/feedback selection
SI	scan input
SO	scan output

### VI. 시뮬레이션 및 결과

본 논문에서 제안한 BIST 실현방법 중 그림 6과 그림 7의 두 알고리즘을 C-언어로 실현하여 주어진 회로에 대한 입력 그룹화와 테스트 포인트 결정과정을 자동화하였다. 제안한 방법을 실행할 대상회로는, 먼저 VHDL로 설계한 후 Synopsys의 합성 툴을 사용하여 합성한 결과를 사용하였다. 본 논문에서의 대상고장은 고착고장(stuck-at fault)으로 선택하였으며, 회로내의 모든 노드에서 고착-0 고장과 고착-1 고장을 가정하였다(중복고장을 제거(fault collapsing)하지 않았음).

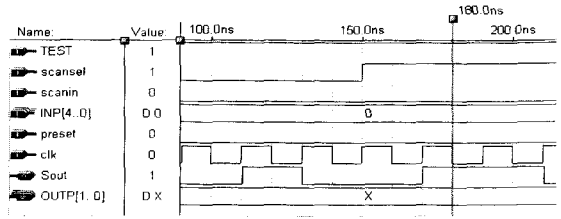
먼저, 본 논문에서 제안한 BIST를 구현한 회로가 의도대로 동작하는 지를 실험하였다. 대상회로는 그림 4



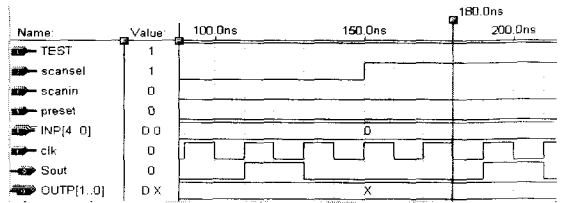
의 회로이며, 그림 9와 같이 BIST가 실현된 회로에 대해 시뮬레이션을 수행한 결과를 그림 10에 나타내었다. 그림 10에서 scanin은 그림의 SI, Sout은 그림 9의 SO에 해당한다. 테스트 수행시 테스트 입력이 모두 인가된 후 압축된 테스트 결과는 SO를 통해 주사출력하여 고장유무를 판별하므로, 시뮬레이션에서는 Sout의 결과가 정상 동작과 일치하는 지를 검증하여야 한다. 그림 10의 시뮬레이션 결과 (b)의 GI 출력에 고착-1 고장이 발생하였을 때의 결과가 (a)의 고장이 발생하지 않았을 때의 결과와 180ns에서 논리값의 차이를 보여 고장이 검출됨을 알 수 있다.

그림 10 회로와 여러 가지 대상회로를 통해 본 논문에서 제안된 방법이 테스트 시간을 얼마나 감소시키는 지를 시뮬레이션을 통해 실험하였으며, 그 결과는 표 2에 나타내었다. 이 표에서 C499회로는 기 생성된 고장 목록을 그대로 사용하였는데, 분석결과 이 목록은 중복 고장이 모두 제거된 것으로 확인되었으나, 그 외의 회로는 중복고장을 제거하지 않았다. 표에서 보는 바와 같이 실험대상은 테스트 시간과 고장검출율이었으며, 테스트 시간은 TPG가 테스트 패턴을 생성하는 기준 클럭수로 정의하였다. 표에서 보는 바와 같이 제안한 방법은 테스트 시간에 있어서 매우 큰 감소를 가져올 수 있으며, 고장검출률면에서도 가정한 모든 고장을 검출할 수 있었다. 따라서 본 논문에서 제안한 방법은 최근 BIST의 가장 큰 문제점으로 대두되고 있는 테스트

시간의 감소 및 고장검출률 증가에 매우 효과적이라 할 수 있다.



(a)



(b)

그림 10. 그림 9의 회로에 대한 시뮬레이션 결과  
(a) 고장이 없는 경우, (b) 고장이 발생한 경우

Fig. 10. Simulation result for the circuit in Fig. 9.  
(a) Fault-free case, (b) Faulty case

Ⅶ. 결 론

본 논문에서는 BIST 수행을 위한 테스트 시간을 감소시키기 위한 BIST 실현방법을 제안하였다. 이 방법은 회로내의 깊이 1인 점들을 기준으로 PI들의 독립성을

표 2. 여러 대상회로에 제안한 방법을 적용한 실험 결과

Table 2. Experimental result by applying the proposed BIST for various CUT.

CUT	#of TPs	Test time		H / O		Rel. H/O (%)		Fault Coverage(%)
		PRP	Propose	PRP	this	Org.	PRP	
16-bit-Adder	4	$\sim 2^{32}$	$\sim 2^{16}$	80	72	83.72	-10.00	100
8-bit-comparator	8	$\sim 2^{19}$	$\sim 2^{11}$	39	47	82.45	20.51	100
Binary decoder with enable	2	$\sim 2^9$	$\sim 2^7$	24	26	162.50	8.33	100
8-bit ALU	12	$\sim 2^{20}$	$\sim 2^{12}$	44	60	55.04	36.36	100
32-bit parity generator	4	$\sim 2^{32}$	$\sim 2^9$	65	50	277.78	-23.08	100
C17	2	$\sim 2^5$	$\sim 2^3$	12	14	233.33	16.60	100
C432	11	$\sim 2^{36}$	$\sim 2^{18}$	79	83	51.90	5.06	100
C499	12	$\sim 2^{41}$	$\sim 2^{16}$	114	113	55.90	-0.87	100
C880	12	$\sim 2^{60}$	$\sim 2^{20}$	146	130	33.94	-10.96	100
C1356	10	$\sim 2^{41}$	$\sim 2^{14}$	114	107	19.59	-6.14	100
C1908	10	$\sim 2^{31}$	$\sim 2^{17}$	91	95	10.79	4.39	100
C3540	13	$\sim 2^{50}$	$\sim 2^{18}$	122	116	6.95	-4.91	100

고려하여 그룹화하고, 기준점들의 특성에 따라 테스트 포인트를 삽입하는 방법을 사용한다. 테스트 포인트는 기존의 제어성/관찰성이 떨어지는 지점이 아닌 새로운 정의에 의해 결정되었으며, 그 기준은 두 개 이상의 기준점이 병합하는 점을 기초로 하였다. 또한 테스트 포인트들과 PO들의 테스트 응답 압축기를 결합하여 고장 검출률을 높이도록 하였다.

제안된 방법은 여러 가지 대상회로에 대해 실험을 통해 테스트 시간의 감소 및 고장검출률의 증가를 확인하였으며, 이 때 사용된 고장의 종류는 회로내의 모든 노드들에 대한 고착-0 및 고착-1 고장이었다. 실험 결과 최고  $1/2^{40}$  의 테스트 시간 감소를 얻을 수 있었으며, 그 외의 회로에서도 상당한 테스트 시간이 감소하였다. 고장검출률에 있어서도 제안한 방법은 100%의 고장검출률을 보여 기존의 방법에 비해 고장검출률면에서도 우수함을 보였다.

따라서 본 논문에서 제안한 방법은 최근의 BIST에 대한 문제점으로 대두되고 있는 테스트 시간의 감소와 고장검출률의 증가를 위해 효과적으로 사용될 수 있을 것으로 기대된다. 본 논문에서 제안한 방법은 삽입되는 테스트 포인트수에 따라 부가 하드웨어의 양이 결정되며,  $n$ 개의 입력과  $m$ 개의 테스트 포인트, 그리고  $k$ 개의 출력력을 가진 회로에 대해  $(n+m+k)$ 개의 플립-플롭과  $(n+m+1)$ 개의  $2 \times 1$  MUX, 그리고  $(m+k)$ 개의 AND 게이트들이 테스트를 위해 부가된다. 또한 이 회로에 의한 정상동작시의 지연시간은 두 개의 MUX와 두 개의 AND 게이트를 통과하는 시간이다. 이 부가회로의 양은 회로가 커질수록 상대적으로 작아지며 지연시간 또한 VLSI와 같은 대형회로에서는 무시할 수 있을 정도로 작은 값이라 할 수 있으므로, 본 논문에서 제안한 BIST 실현방법은 VLSI와 같은 대형회로에서 더욱 큰 효과를 얻을 수 있으리라 사료된다.

## 참 고 문 헌

- [1] Jon Turino, "Test Economics in the 21st Century", IEEE Design & Test, pp. 41-44, July-Sep. 1997.
- [2] Gadi Singer, "VTS 97 Keynote: The future of Test and DFT", IEEE Design & Test, pp. 11-14, July-Sep. 1997.
- [3] Michael J. Riezenman, "Technology 1998, Test & Measurement", IEEE Spectrum, pp.65-69, Jan. 1998.
- [4] Kevin O'Leary, "설계와 제조의 통합", Electronic Engineering, pp. 28-30, July 1998.
- [5] A. Ahmad, "Achievement of Higher Testability Goals through the Modification of Shift Registers in LFSR-based Testing", Int. J. Electronics, Vol. 82, No. 3, pp. 249-260, 1997.
- [6] Thyagaraju Damarla and Avinash Sathaye, "Applications of One Dimensional Cellular Automata and Linear Feedback Shift Registers for Pseudo-Exhaustive Testing", IEEE Trans. on CAD, Vol. 12, No. 10, pp. 1580-1591, Oct. 1993.
- [7] Chin-Ang Chen, Sandeep K.Gupta "BIST test pattern generators for two-pattern testing-theory and design algorithms" IEEE transaction on computers, vol.45, No.3, pp. 257-268 March 1996.
- [8] Chih -Ang Chen, Chen Sandeep k.Gupta. "Design of efficient BIST test pattern generation for delay faults" IEEE Transaction on Computer-Aided design of Integrated Circuits and systems. vol.15, No.12, pp 1568 - 1575, 1996.
- [9] Nur A, Touba and Edward J, McClusky, "Test point Insertion for Non-feedback Bridging faults" CSL-TR-96-703, 1996.
- [10] Brend Konneman, Joachim Mucha and Gunther Zwiehoff, "Built-In Logic Block Observation Techniques", IEEE Test Conference, pp. 37-41, 1979.
- [11] Vishwani D. Agrawal, Charles R. Kimw, and Kewal K. Saluja, "A Tutorial on Built-In Self Test, Part 2: Application", IEEE Design & Test, pp. 69-77, June 1993.
- [12] Vishwani D. Agrawal, Charles R. Kimw, and Kewal K. Saluja, "A Tutorial on Built-In Self Test, Part 1: Principles", IEEE Design & Test, pp. 73-82, Match 1993.
- [13] Solomon W. Golomb, Shift Register Sequences,

- Aegean Park Press, Laguna Hills, CA, 1982.
- [14] Jacob Savir, "Shrinking Wide compressors", IEEE transaction on Computer-Aided design of integrated circuit and systems. Vol. 14, No. 11, pp. 1379-1387, November 1995.
- [15] Nur A. Toubia, "Synthesis Techniques for Pseudo-Random Built-In Self-Test", A dissertation submitted to the Dept. Electrical Eng. of Stanford University, 1996.
- [16] Chung-Hsing Chen and Daniel G. Saab, "A Novel Behavioral Testability Measure", IEEE Trans. CAD, Vol. 12, No. 12, pp. 1960-1970, Dec. 1993.

---

 저 자 소 개
 

---

## 張允碩(學生會員)

1969년 4월 22일생. 1997년과 1999년 8월에 광운대학교 전자재료 공학과에서 각각 학사와 석사학위를 취득하였으며, 99년 8월부터 현재까지 (주)아이앤씨 테크놀러지에서 ASIC설계팀 연구원으로 근무중이다. <주관심 분야 : VLSI 설계, CAD, Design-for-Testability 등>



## 金東郁(正會員)

1960년 8월 23일생. 1983년과 1985년 2월, 한양대학교 전자공학과에서 각 1991년 미국 Georgia Institute of Technology의 전기공학각 학사와 석사학위를 취득하였으며 과에서 박사학위를 취득하였다. 1992년부터 현재까지 광운대학교 전자재료공학과와 부교수로 재직중이며 현재 Professional activities of SSCS/EDS Joint chapter of IEEE Seoul section의 chief로 활동하고 있다. <주관심 분야 : VLSI 설계, CAD, Digital Testability 등>