

분할 및 병렬 처리 방법에 의한 BIST의 테스트 시간 감소

論 文
49D-6-5

Test Time Reduction for BIST by Parallel Divide-and-Conquer Method

崔炳九* · 金東郁**

(Byung-Gu Choi · Dong-Wook Kim)

Abstract - BIST(Built-in Self Test) has been considered as the most promising DFT(design-for-test)scheme for the present and future test strategy. The most serious problem in applying BIST(Built-in Self Test) into a large circuit is the excessive increase in test time. This paper is focused on this problem. We proposed a new BIST construction scheme which uses a parallel divide-and-conquer method. The circuit division is performed with respect to some internal nodes called *test points*. The test points are selected by considering the nodal connectivity of the circuit rather than the testability of each node. The test patterns are generated by only one linear feedback shift register(LFSR) and they are shared by all the divided circuits. Thus, the test for each divided circuit is performed in parallel. Test responses are collected from the test point as well as the primary outputs. Even though the divide-and-conquer scheme is used and test patterns are generated in one LFSR, the proposed scheme does not lose its pseudo-exhaustive property. We proposed a selection procedure to find the test points and it was implemented with C/C++ language. Several example circuits were applied to this procedure and the results showed that test time was reduced upto $1/2^{151}$ but the increase in the hardware overhead or the delay increase was not much high. Because the proposed scheme showed a tendency that the increasing rates in hardware overhead and delay overhead were less than that in test time reduction as the size of circuit increases, it is expected to be used efficiently for large circuits as VLSI and ULSI.

Key Words : BIST, LFSR, Test Point, Pseudo-Random Patterns, Input Grouping

1. 서 론

최근의 급격한 집적도 증가로 IC의 면적과 속도의 비용이 급속히 감소하고 있다[1]. 그러나 이로 인해 설계와 공정 상에서 해결되어야 할 많은 문제가 발생하는데, 그 중 가장 시급히 해결하여야 할 문제로 테스트 문제가 대두되고 있다[2]. 현재의 경향은 설계과정에서부터 테스트 과정을 고려하는 것인데, 이를 DFT(Design-for-Test)라 부른다. DFT방법 중에서 BIST는 현재는 물론 향후에도 가장 효율적인 테스트 전략으로 인식되고 있는데, 이는 자동 테스트와 정상동작 속도에서의 테스트가능성 때문이다[2,3]. 일반적으로 테스트 가능성의 증가는 다음 세 가지를 희생시키는데, 정상 동작속도의 저하, 부가하드웨어의 증가, 전력소비의 증가가 그들이다. 그러나 이러한 문제들은 집적도의 증가에 따라 어느 정도 극복될 수 있으므로[4], 본 논문에서는 이러한 문제를 깊이 다루지는 않는다. 반면, 테스트 시간은 입력 수에 따라 지수 함수

적으로 증가하는데, 이것이 BIST 적용시 가장 큰 문제점으로 지적되고 있다. BIST에서는 기본적으로 회로의 입력수가 n 일 때 2^n-1 개의 CRC(Cyclic Redundant Code)로 구성된 의사-무작위패턴(Pseudo-Random-Pattern, PRP¹⁾)이 가장 많이 사용된다. 예를 들어 50개의 입력을 가진 회로를 100Mhz의 테스트 클럭으로 검사를 할 경우, 검사시간은 약 4.3개월이 소요되는데, 이는 실용 불가능한 수치이다.

선형피드백 쉬프트 레지스터(Linear Feedback Shift Register, LFSR)를 사용하는 BIST의 개념이 발표된 이래로 [5], BIST 수행시간의 감소를 위해 많은 연구가 진행되어 왔다[6-11]. [8]에서는 회로를 두 개 이상의 부회로로 분할하여 BIST를 적용하는 방법이 제안되었는데, 이 방법은 나누어진 각각의 부회로들과 주입력(Primary Input, PI)/주출력(Primary Output, PO)사이에 많은 연결배선으로 구현되므로, 실제적인 테스트 시간감소 효과가 크게 나타나지 않는 경우가 많다. BIST를 각 모듈단위로 적용하는 방법이 [9]에서 제

1) 의사-무작위 패턴 중 가장 널리 사용되는 패턴 세트는 최대 길이를 갖는 PRP이며, 그 내용 측면에서 보면 모든 가능한 패턴(전경우 패턴)에서 모든 비트가 '0'인 패턴만이 제외된다. 따라서 이런 패턴을 의사-전경우(pseudo-exhaustive) 패턴이라고도 한다. 본 논문에서는 의사-무작위 패턴(MLPRP, Maximum Length Pseudo-Random Pattern)과 의사-전경우 패턴을 공히 이 패턴 세트를 가리키는 용어로 사용한다.

* 準 會 員 : 光云大 電子材料工學科 碩士課程

** 正 會 員 : 光云大 電子材料工學科 副教授 · 工博

接受日字 : 2000年 1月 29日

最終完了 : 2000年 4月 29日

안되었는데, 이것은 여러 모듈들이 버스 구조로 연결되어 있는 시스템 IC에서의 적용에 초점을 맞추고 있다. 1997년에는 다중 시드(seed)를 통한 순차길이를 감소시키는 방법이 제안되었는데[10], 다중 시드의 동작을 위해서 프로그램가능한 LFSR을 사용하여 PRP에 비해 1/4 정도로 테스트 길이를 줄일 수 있었다. [11]에서는 입력수 감소를 통한 테스트 시간 감소 방법이 제안되었는데, 이 방법은 PO에 연계된 PI들을 분할하고, 테스트 패턴들을 적절히 구성하여 테스트시간을 감소시켰다. 실험결과에 의하면 최대 45%까지 테스트시간 감소 효과를 얻었다. 그러나 이 방법을 적용할 경우 패턴을 재구성하기 위하여 많은 계산시간이 소요되는 단점을 가지고 있다.

본 논문에서는 테스트 시간을 감소시키는 새로운 방법을 제안하고자 한다. 이를 위해 테스트 포인트(Test Point, TP)들을 새로이 정의하고 이와 연계되어 PI들을 그룹화하여 주어진 회로를 분할한다. 일반적으로 TP는 회로 내에서 테스트 가능성을 높일 수 있는 노드로 정의되며[12], RTL에서도 유사한 정의가 사용된다[13]. 그러나 본 논문에서는 회로의 연결도를 고려하여 TP들을 선정한다. 본 논문에서는 LFSR을 통해 테스트 패턴을 생성하고 MISR(Multiple Input Signature Register)을 통해 테스트 응답을 처리하며 테스트 패턴으로는 PRP가 사용되는 BIST를 수행하는 것으로 가정한다.

본 논문의 구성은 다음 장에서 본 논문의 회로 분할 원리에 대하여 간단한 설명을 하고, III장에서 본 논문에서 사용하는 용어들에 대해 정의한다. IV장에서는 TP를 선정하기 위한 조건들을 설명하고, V장에서 TP들을 선정하는 방법 및 절차에 대하여 설명한다. VI장에서는 TP를 이용하여 BIST를 구현하기 위한 부가 하드웨어에 대해 설명하고, VII장의 실험 결과와 고찰을 통하여 VIII장에서 결론을 맺는다.

2. 내부노드에 대한 회로분할 및 입력그룹화

본 장에서는 그림 1의 예제회로를 통하여 본 논문에서 제안하는 기본적인 회로분할 방법의 개요에 대하여 간략히 설명하고자 한다.

그림 1의 (a)는 ISCAS'85의 C17회로이며, 5개의 PI와 2개의 PO를 가지고 있다. 이 회로를 PRP를 사용한 BIST로 테스트를 할 경우 테스트 클럭은 약 2⁵개가 필요하게 된다. 다

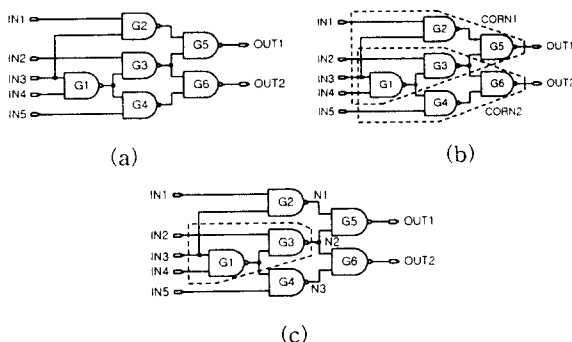


그림 1 ISCAS'85의 C17 회로에 대한 회로분할
Fig. 1 Circuit division for C17 circuit from ISCAS'85

지될 회로에서는 일반적으로 임의의 PO에 대하여 모든 PI들이 영향을 미치는 것이 아니므로, PI들은 PO와의 관계에 따라 나누어질 수 있다[11].

그림 1의 (b)에서 PO에 대하여 영향을 미치는 PI집합을 표시하면 다음과 같다.

$$\begin{aligned} OUT1 &= f(IN1, IN2, IN3, IN4) \\ OUT2 &= f(IN2, IN3, IN4, IN5) \end{aligned} \quad (1)$$

PO에 대한 함수관계에서 보듯이, PO에 대하여 PI들은 분리되므로 두 그룹을 동시에 고려할 필요가 없다. 따라서 LFSR에서 생성된 패턴을 적절히 할당하면 테스트 클럭은 약 2⁴개로 테스트가 가능하게 된다. 이러한 개념은 회로의 내부노드에 대하여도 확장해서 적용할 수 있는데, 그림 2 (c)에서 보여주듯이 내부노드 N2에 대하여 PI는 {IN2, IN3, IN4}가 영향을 미치고 있으며, 이러한 관계를 이용할 경우 테스트시간을 더욱 감소시킬 수 있다. 그림 1 (a) 회로의 내부노드의 PI에 대한 관계는,

$$\begin{aligned} M &= f(IN1, IN3) \\ N2 &= f(IN2, IN3, IN4) \\ N3 &= f(IN3, IN4, IN5) \end{aligned} \quad (2)$$

로 나타낼 수 있으며 PO에 대한 함수는 다음과 같다

$$\begin{aligned} OUT1 &= \{M, N2\} \\ OUT2 &= \{N2, N3\} \end{aligned} \quad (3)$$

이 경우 적절히 LFSR 비트들을 할당하면 테스트 시간²⁾은 약 2³개로 감소시킬 수 있다. 이러한 테스트시간 감소를 위해서 내부노드 이전 단계에 대한 테스트응답과 이후 단계에 대한 테스트패턴 인가의 기능을 내부 노드가 모두 수행하여야 하는데, 그림 1의 경우는 식 (2)에서와 같이 세 개의 노드가 필요하게 된다. 이 경우 이 세 개의 노드에 부가되는 하드웨어로 인하여 전체적으로 부가하드웨어와 정상동작시의 속도 감소가 발생하게 된다. 본 논문의 의도는 높은 테스트시간 감소효과를 얻는 동시에 부가하드웨어의 양을 최소로 하는데 있다. 따라서 본 논문에서는 최소한의 내부노드를 찾는 데 주안점을 둔다.

이러한 관점에서 보면, 그림 1 (c)에서 N3는 기준점으로 선정되지 않아도 무방하다. 그 이유는 이미 N2에 대한 PI 그룹이 세 개의 입력을 가지고 있으며, MLPRP생성을 위한 LFSR의 비트 수는 단지 3비트이면 만족되기 때문이다. LFSR의 비트를 각 PI와 기준점에 할당할 때 의사-전경우의 성질을 유지하면서 IN2와 IN5를 같은 비트에 할당할 수 있으므로 PI 그룹은 단지 N1, N2의 두 개만 필요하고, N3에 대하여는 기준점으로 선정할 필요가 없다. N1, N2를 통해 주어진 회로는 다음과 같이 4개의 부회로로 나누어진다.

$$\begin{aligned} M &= f(IN1, IN3) \\ N2 &= f(IN2, IN3, IN4) \\ OUT1 &= f(M, N2) \\ OUT2 &= f(IN5, N2) \end{aligned} \quad (4)$$

앞에서 간단히 언급했듯이, 본 논문에서의 기준점(위의 예제에서 N1, N2)은 테스트 응답정보의 검출과 테스트패턴 인

2) 본 논문에서 테스트시간은 테스트를 수행하는데 필요한 테스트 클럭(test clock)수로 나타낸다.

가의 기능을 모두 수행한다. 테스트패턴을 생성할 때는, 기준점 이후 회로에 대하여 별도의 패턴을 생성할 필요가 없다. 대신 그림 2와 같이 기준점 전단의 LFSR 비트들을 공유할 수 있다. 그러나, 이 경우 패턴세트가 의사-전경우의 성질을 상실하지 않으려면 기준점 선정과 비트할당 시 공유되는 비트의 선정에 유의해야 한다. 그림 2를 예로 들어 설명하면, LFSR의 동일 비트를 IN2와 N2가 공유한다면 G6의 두 입력은 의사-경우의 성질을 상실한다. 따라서 IN2에 할당된 비트는 N2와 공유할 수 없다. 표 1에 그림 2의 회로에 대해 의사-전경우의 성질을 유지할 수 있는 LFSR 비트의 할당을 보여주고 있다. 여기서 ‘-’는 비트를 할당하지 않음을 나타낸다.

한가지 더 언급할 것은, 테스트 응답기의 비트 수는 두 개의 PO와 두 개의 기준점으로 구성되어 총 4 비트를 가진다. 그리고, 고장 검출률(Fault Coverage, FC)은 다음의 식으로 표현되므로

$$FC_{n-bit MISR} \cong 1 - \frac{1}{2^n - 1} \quad (5)$$

이 방법은 고장 검출률을 향상시키는 부가적인 효과도 얻을 수 있다.

3. 정 의

제한한 방법에 대한 설명을 진행하기 전에, 본 논문에서 사용되는 몇 가지 용어들을 정의한다.

<정의 1> 논리회로에서 주어진 PI(IN_j)에 대한 특정 내부노드(N_i)의 깊이(Depth_{IN_j}(N_i))는 IN_j에서 N_i까지 통과하여야 하는 게이트 수로 정의한다. 즉,

$$Depth_{IN_j}(N_i) \equiv \text{Number of gates between } IN_j \text{ and } N_i \quad (6)$$

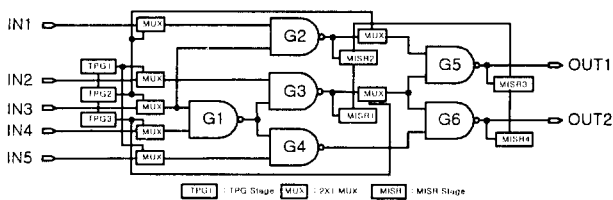


그림 2 TP들을 이용한 C17의 BIST 구현
Fig. 2 BIST scheme of C17 using TPs

표 1 그림 1 회로에 대한 LFSR 비트 할당
Table 1 LFSR bit assignment for the circuit in Fig.1

TP	Node	bit1	bit2	bit3
iTP	N2	IN2	IN3	IN4
aTP	N1	IN1	IN3	
PO	OUT1	-	IN1	N2
PO	OUT2	-	IN5	N2

<정의 2> 노드 N_i의 최소깊이(MinDepth(N_i))는 각 PI에서 N_i까지의 깊이 중 최소값으로 정의한다. 즉,

$$MinDepth(N_i) \equiv \min \{ Depth_{IN_j}(N_i) \text{ for } 1 \leq j \leq n, n = \text{number of PIs} \} \quad (7)$$

<정의 3> 노드 N의 PI 그룹(PIG(N))은 노드 N에 영향을 미치는 PI들의 집합으로 정의한다. 또한 PIG(N)의 크기는 PIG(N)에 속한 PI들의 수로 정의하며, #PIG(N)으로 표현된다.

<정의 4> 노드 N의 값에 영향을 미치는 각각의 PI들이 또한 노드 M의 값에도 영향을 미치면 노드 N은 노드 M에 포함(N ⊆ M)되었다고 말한다. 즉,

$$N \subseteq M \text{ if } PIG(N) \subseteq PIG(M) \quad (8)$$

그림 1의 예제회로에서 Depth_{IN2}(N2)=1이나 Depth_{IN3}(N2)=2이다. 따라서 MinDepth(N2)=1이다. PIG(M)={IN3, IN4}이므로 #PIG(M)=2이고 PIG(N2)={IN2, IN3, IN4} 즉, #PIG(N2)=3이므로 M ⊆ N2이고 M ⊆ N3이다.

<정의 5> 테스트 포인트(TP)는 입력분할 기준점의 기능을 하는 내부노드로 정의한다.

그림 1, 2의 예제회로에서 위 <정의 5>에 의해 TP로 선정될 노드는 N1과 N2이다.

4. 테스트 포인트의 선정 조건

앞장에서 언급하였듯이, TP로 정의된 내부노드들은 테스트 수행을 위한 입/출력의 기능을 모두 수행하여야 한다. 즉, 회로의 PI로부터 TP까지의 테스트 결과를 추출하고, TP로부터 PO에 이르는 회로에 테스트 벡터를 인가하는 역할을 한다. 이 같은 구성은 기존의 연구들에서 사용했던 내부노드의 용도와 유사하나, 구체적인 용도와 선정의 특징에 있어서는 확연한 차이를 보인다. 첫째, 본 논문에서는 PI 그룹화에 의해 나누어지는 몇 개의 부회로들이 상호간의 독립성을 갖는다. 이 같은 특성을 통해 MLPRP를 생성하면서 동시에 LFSR의 비트 수를 줄일 수 있게 된다. 내부 노드에 대한 테스트 패턴은 새로이 생성되는 것이 아니고, PI 그룹에 할당된 패턴을 공유한다. 마지막으로 TP들은 검사 정보를 PO와의 조합에 의해 하나의 MISR을 형성하여 추출함으로써 고장 검출률을 향상시키는 부수 효과를 얻는다.

기본적으로, PI 그룹들 중에서 그 크기가 가장 큰 그룹이 가장 중요한 역할을 하는데, 그 이유는 이 그룹의 크기가 테스트 시간을 결정하기 때문이다. 본 연구의 목적이 테스트 시간을 감소시키는 것에 있고 이것이 가장 큰 PI 그룹의 크기에 달려있기 때문에, 가장 큰 입력그룹의 크기는 테스트 시간을 충분히 감소시킬 수 있을 정도로 작아야 할 것이다. 테스트시간 감소율(Test Time Reduction Rate, TTRR)은 다음과 같이 단순히 표시할 수 있다.

$$TTRR = \frac{\text{number of test clocks after input grouping}}{\text{number of test clocks for PRP}} \approx \frac{2^m - 1}{2^n - 1} \approx 2^{m-n} \quad (9)$$

여기서 n은 PI의 수, m은 최대 PIG의 크기를 각각 나타낸다.

그룹화된 후에는 단지 하나의 LFSR에 의해 테스트 패턴이 생성되며 이 패턴들은 모든 PIG가 공유한다. 이로써 모든 PIG들에 대하여 MLPRP가 인가되는데, 그 이유는 가장 큰

PIG에 기준을 두어 패턴이 생성되기 때문이다. 이 같은 성질은 다음의 정리로부터 얻어졌다.

[정리 1] 2^n-1 개의 n 비트 MLPRP 세트에서 임의로 선택한 $k(k \leq n)$ 비트 2^n-1 패턴 세트는 k 비트 MLPRP를 포함한다.

위 정리 1의 증명은 여러 문헌을 통해 많은 설명이 발표되었으므로[14,15], 본 논문에서는 생략하기로 한다.

그림 2의 예제에서 PI를 그룹화하기 위해 필요한 TP들로 인해 과도하게 많은 부가하드웨어가 요구되었다. 이 예에서 보듯이, 부가되는 하드웨어의 면적이나, 전력소모, 정상동작시의 지연 등의 문제로 인해 많은 수의 노드들을 TP로 선정하는 것은 비효율적이거나 허용되지 않는 경우가 많다. 사실, TP의 개수는 테스트 시간의 감소와 부가 하드웨어 증가간에 상충적인 관계를 갖고 있다. 즉, 많은 수의 테스트 포인트가 사용될수록 테스트 시간 감소율은 높아지지만, 이에 반해 속도저하와 부가하드웨어의 면적은 증가하게 되는 것이다. 이러한 이유에서 TP를 선정하는 데는 엄밀한 기준이 필요하게 되며, 이러한 목적을 위해 다음 세 가지의 조건이 만족되어야 한다.

첫째로, 임의의 회로에 대한 테스트 시간은 가장 큰 PIG의 크기에 의존한다. 결국 만족할 만한 테스트 시간 감소를 얻으려면 이 PIG의 크기가 충분히 작아야 한다. 만약 충분히 작지 않은 노드가 선택이 된다면 적절한 내부노드를 찾을 때까지 회로를 탐색하는 과정이 필요하다.

MLPRP를 통한 BIST는 거의 전경우 테스트에 근사하며, 모든 비트들이 '0'인 경우를 제외한 모든 패턴이 포함된다. 만약 MLPRP 테스트 특성을 상실하게 되면 고장 검출률이 감소할 수 있다. 본 논문에서 제시하는 방법이 이 MLPRP 특성을 유지하는 것은 다음의 정의와 정리로 설명한다.

<정의 6> TP 선정을 통한 PI 그룹화에 의해 형성된 부회로들에 MLPRP패턴이 인가될 때, 각 부회로들은 서로 '독립적이다'라고 한다.

[정리 2] TP 선정을 통한 PI 그룹화에 의해 형성된 부회로들 중에서 가장 큰 PIG로부터 생성된 MLPRP 패턴이 공유되면, 각 부회로들은 서로 독립적이다.

(증명) PI 그룹화에 의해 형성된 부회로는 그림 3에 보인 것처럼 3가지의 유형으로 분류할 수 있다. 각각의 그림에서 두 개의 부회로만으로 설명을 진행하는데, 두 개 이상의 부회로를 가지는 경우에는 두 개의 부회로씩 거듭하여 고려하면 되므로, 두 개의 부회로에 대한 설명으로 충분하다. 그림 3 (a)는 두 개의 서로 분리된 부회로를 가지고 있다. 정리 1에 의해 이 경우에는 더 큰 PIG에 대해 생성되는 MLPRP를 작은 PIG와 공유함으로써 두 부회로 모두에 대해 MLPRP 테스트가 가능하다. 그림 3 (b)의 경우는 적어도 하나의 입력이 두 부회로에 공통적으로 포함된 경우이다. 그림 2와 같이 패턴이 공유된다고 가정하면, 공통되는 입력의 패턴 비트는 같은 LFSR 비트를 할당하도록 한다. 즉, PI3에 PRP 비트 j 를 할당했다면, 마찬가지로 B회로의 경우도 비트 j 를 PI3에 할당한다. 전체회로에 대해 정상동작이 수행될 때, 즉 PI에 따른 분

리를 하지 않았을 때, 포함된 부회로의 입력은 두 부회로에서 언제나 공통 입력이 된다. 그림 2에서 부회로 A와 B에 대한 테스트응답이 TP1과 TP2로부터 검출되기 때문에 TP1과 TP2는 각 부회로의 출력노드 역할을 하게 된다. 따라서 만약 두 부회로에 공유되는 입력에 동일한 LFSR 비트를 할당하면 두 회로는 각각 MLPRP로의 테스트가 가능해지므로 서로 독립성을 가지게 된다. 마지막으로 그림 3 (c)와 같이 어느 한 회로가 다른 하나 또는 그 이상의 부회로를 포함한 경우를 고찰해보자. 이 경우 B회로가 A회로와 MLPRP를 공유한다는 것은 명확하다. 그림 2의 회로에서 보듯이 B회로의 응답이 아닌 새로운 패턴이 그림 3 (c)의 TP2로 인가된다. B회로와 TP2에 입력 PI1과 PI2의 패턴이 포함되지 않으며, 이 패턴들이 하나의 LFSR로부터 생성되므로 모두가 PRP의 성질을 가지게 된다. 따라서 이 경우도 두 개의 부회로들은 서로 독립적이다. 결과적으로 그림 3의 세 경우 모두 서로 독립성을 유지한다. Q.E.D.

이 정리에 따라 본 논문의 의도에 맞게 부분회로를 선택할 수 있다. 이것을 부분회로의 독립성 또는 간단히 독립성이라고 부르며, 이것이 두 번째 조건에 해당한다.

TP선정에 있어서의 세 번째 조건은 다음과 같다. 제안한 방법에서 모든 테스트패턴의 인가는 입력 그룹화에 의해 이루어진다. 구조적으로 볼 때, 부분회로는 PI와 TP를 입력으로 하고 TP와 PO를 출력으로 하는 콘(cone)을 형성한다. 콘 내부의 입력은 그 콘 안에서만 영향을 미친다. 따라서 전체

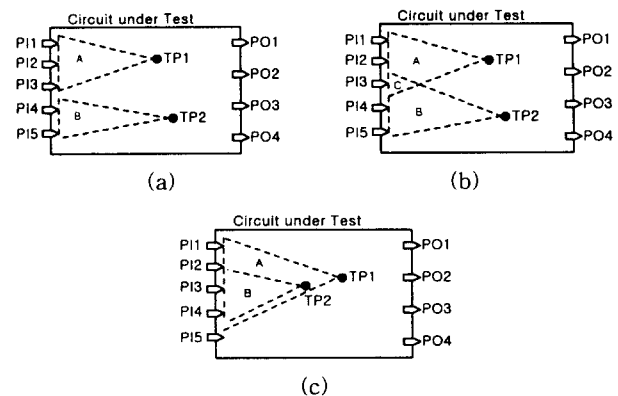


그림 3 두 개의 부회로의 세 가지 유형
Fig. 3 Three configuration classes of two sub-circuits

회로의 노드를 검사하기 위해서는 회로내의 모든 노드들은 적어도 하나 이상의 부분회로에 포함되어야만 한다. 이것을 노드의 부분회로에의 포함성 또는 간단히 포함성이라고 한다.

따라서, TP의 집합에 근거한 부분회로의 집합은 반드시 독립성과 포함성을 만족해야 한다. 또한 최대 PIG의 크기는 적절한 테스트 시간 감소효과를 얻도록 충분히 작아야 한다.

5. 테스트 포인트 선정

5.1 테스트 포인트의 종류

본 논문에서 PI 그룹화를 위한 TP는 초기 TP(initial Test

Point, *iTP*)와 추가 TP(Additional Test Point, *aTP*)로 나누어진다. 회로 내부의 특정 노드가 *iTP*로 선정이 됨으로써 TP 선정 과정이 시작된다. *iTP*의 선정 기준은 다음과 같다.

$$iTP = \{ \text{node which is not PO} \mid (\text{MinDepth}=1) \text{ AND } (\#PIG \text{ is maximal}) \} \quad (10)$$

위 식에서 *MinDepth*=1은 본 논문의 방법을 구현하기 위해 임의로 결정한 기준이다. #PIG가 최대인 조건은 추가되는 *aTP*의 수를 줄이기 위함이며, 이는 다음 단락에서 설명하도록 한다. 특정 회로에서는 식(10)을 만족하는 노드가 여러 개일 수 있는데, 이 경우 식(10)을 만족하는 모든 노드가 *iTP*로 선정된다. 또한 식(10)을 만족하지만 그 크기가 충분히 작지 않은 노드가 선정될 수도 있다. 이 경우에 테스트 시간은 미리 결정된 수준 이내로 제한하여야 하는데, 그 범위 내에서 적당한 #PIG(*k*)를 가지는 노드를 *iTP*로 선정할 수 있다. 이 경우 식(10)은 다음과 같이 변경된다.

$$iTP = \{ \text{node which is not PO} \mid \#PIG \geq k \} \quad (11)$$

또 다른 TP 유형은 *aTP*이다. *aTP*의 목적은 모든 TP가 선정되었을 때, 형성되는 부회로들에 독립성과 포함성이 확보되도록 하는 데 있다. 즉, *iTP*를 통해 부분회로 콘이 형성되면, 나머지 회로에 대해서는 *aTP*를 통해 추가적으로 분해가 이루어지는 것이다. *aTP*의 선정 시에도 *iTP* 선정 시와 같이 독립성과 포함성을 유지하는 범위에서 가장 큰 노드를 선택한다.

5.2 테스트 포인트 선정 절차

TP의 선정 절차를 그림 4에 간단히 나타내었다. 이 절차는 크게 *iTP* 선정과정과 *aTP* 선정과정의 두 부분으로 나누어진다. *iTP*는 식(10)의 조건을 만족하는 노드를 선정하며 #PIG가 만족스럽지 못할 때에는 식(11)에 따라 추가적인 검색

```

1: TestPointSelect()
2: Inputs : CUT, input groups, list of # of PIs in
           input groups
3: Outputs : test point set(iTP, aTP)
4: begin
5:   find all nodes satisfying eq. (10);
6:   if their #PIGs are too big, then
7:     select a node with appropriate #PIG
8:   take all as iTPs;
9:   for all iTPs, do
10:    advance one gate toward POs;
11:    for all the input nodes, Nj
12:      if PIG(TP) ≥ PIG(Nj) and satisfy independency
           property and inclusion property then
13:        take Nj as a aTP;
14:    else
15:      backward trace one gate;
16:      goto line 12;
17:    if Nj is PO, then
18:      end procedure;
19:   loop
20: end
    
```

그림 4 TP 선정 절차
Fig. 4 TP selection procedure

작업이 이루어진다. *iTP* 선정이 끝나면 *aTP* 선정 절차에 들어간다. 기본적으로 *iTP* 선정 이후에 *iTP*에 의해 형성되는 콘에 포함되지 아니한 나머지 회로를 고찰하여 *aTP*를 선정하게 된다. 따라서 *aTP* 검색 작업은 각 *iTP*로부터 시작한다. 검색 작업은 출력 쪽으로 한 게이트 단위씩 추적하면서 #PIG 조건, 독립성, 그리고 포함성 조건을 만족하는 노드를 검색해 나가며, PO를 만나면 검색 작업은 종료된다.

5.3 예제

두 개의 예제를 통해 TP 선정 절차에 관하여 좀더 자세한 설명을 하고자 한다. 그림 5에 보이는 첫 번째 예제회로는 2진 해독기(Binary decoder)이다. 이 회로는 9개의 입력과 6개의 출력, 그리고 10개의 내부 노드를 가지고 있다.

식(10)을 만족하는 노드는 N10으로서 #PIG=7이며, 이 크기는 만족스러운 크기로 가정하여 *iTP*로 선정한다. 그 후 *aTP* 선정과정을 시작하는데, 먼저 N10에서 한 게이트를 출력쪽으로 진행하면 *y*[0], *y*[1], *y*[2] 그리고 *y*[3]를 만나게 된다. 그런데 이들이 N1과 N2노드를 포함하고 있으므로 판단은 N1과 N2로부터 다시 시작한다. 이 과정에서 N7, N8, N9 노드를 만나게 된다. PIG(N8) ≤ PIG(N10), PIG(N9) ≤ PIG(N10) 이면서 #PIG(N10)이 테스트 시간의 조건에 만족하므로 N8과 N9는 선정할 필요가 없고 단지 N7만이 *aTP*로 선정된다. 이 추적에서 PO들을 만났으므로 검색 작업을 종료

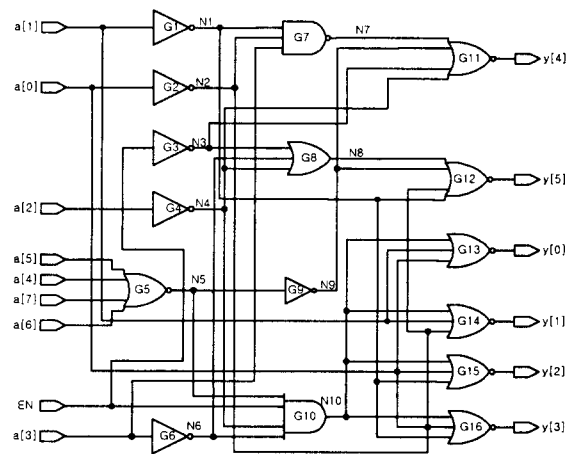


그림 5 여기 신호를 포함한 2진 해독기
Fig. 5 Binary decoder with enable signal

하고 결과적으로 N7과 N10이 TP로 선정된다. 표 2는 이 두 TP에 의한 LFSR 비트들의 할당 결과를 나타내었다.

표 2에서 보듯이 *iTP*인 N10에 의거해서 7비트 LFSR이 구성된다. 주 입력들 중 *a*[3]은 N7과 N10의 PIG에 공통으로 포함되어 있다. 따라서 *a*[3]의 할당은 공히 bit3에 할당하였다. N7과 N10으로 인해 형성된 부분회로들의 검사 응답은 각각 N7과 N10을 통해 검출되며 나머지 회로의 테스트 응답은 PO 단에서 검출된다.

두 번째 예제회로는 그림 6의 32-비트 패리티 생성기이다. 이 회로는 32개의 PI와 1개의 PO, 17개의 내부 노드를 가지고 있다. 먼저 *iTP*로 식(10)을 만족하는 노드로서 N17이 선

정 대상이 된다. 그런데, 만족스러운 테스트시간 감소효과를 얻으려면 대략 #PIG=16 이하의 노드가 필요하다고 한다면 #PIG(N17)=30이므로 부적합하다. 따라서 식 (11)을 적용하여 검색한 결과 #PIG(N10)=16이므로, 최종적으로 N17 대신 N10을 iTP로 선정하게 된다(이 작업의 결과 테스트 길이는 약 2^{16} 으로 감소할 수 있다). 이제 N10으로부터 추적 작업을 시작하면, N12, N14, N16 그리고 N17노드를 만나게 된다.

표 2 그림 5 회로에 대한 LFSR 비트 할당
Table 2 LFSR bit assignment for the circuit in Fig. 5

TP	Node	bit1	bit2	bit3	bit4	bit5	bit6	bit7
iTP	N10	EN	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
aTP	N7	-	-	a[3]	-	-	a[0]	a[1]
PO	y[0]	-	-	-	-	N10	a[0]	a[1]
PO	y[1]	-	-	-	-	N10	a[0]	a[1]
PO	y[2]	-	-	-	-	N10	a[0]	a[1]
PO	y[3]	-	-	-	-	N10	a[0]	a[1]
PO	y[4]	-	-	-	-	N10	a[0]	a[1]
PO	y[5]	-	-	-	-	N10	a[0]	a[1]

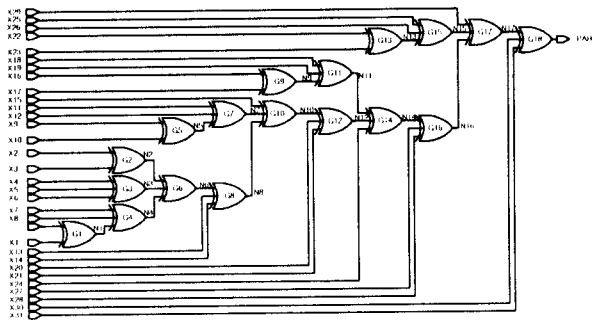


그림 6 32-비트 패리티 생성기
Fig. 6 32-bit parity generator

N10을 한 개의 입력으로 계산하면 $N12 \leq N14 \leq N16 \leq N17$ 이고 #PIG(N12)=3, #PIG(N14)=8, #PIG(N16)=10, #PIG(N17)=15 이므로 단지 N17만이 aTP로 선정된다. 결과적으로 단 2개의 TP만이 선정되며 이 결과 테스트 클럭은 2^{32} 에서 2^{16} 개로 감소된다. LFSR 비트 할당 결과는 다음 표 3과 같다.

6. BIST 부가 하드웨어

이미 언급한 바와 같이, BIST 구현 시 PI 단과 TP 노드에 추가적인 하드웨어가 필요한데, 추가될 회로를 그림 7에 나타내었다. PI에 추가되는 부가회로는 PI 그룹화되지 않은 경우와 동일하다. 'SELECT' 신호는 정상동작과 테스트 동작을 선택하기 위해 부가되었다. TP에 추가될 회로는 2x1 멀티플렉서와 플립-플롭이다. 그림에서 '⊕' 기호는 LFSR 구성을 위한 exclusive-OR를 뜻한다. 그림 7에서 PO 단에 추가될 회로는 나타내지 않았는데, 일반적인 BIST 구성과 유사하며

표 3 그림 6회로에 대한 LFSR 비트 할당
Table 3 LFSR bit assignment for the circuit in Fig. 6

TP	NODE	bit1	bit2	bit3	bit4	bit5	bit6	bit7	bit8	bit9	bit10	bit11	bit12	bit13	bit14	bit15	bit16
iTP	N10	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16
aTP	N17	N10	X17	X18	X19	X20	X21	X22	X23	X24	X25	X26	X27	X28	X29	X30	-
PO	PAR	N17	X31	X32	-	-	-	-	-	-	-	-	-	-	-	-	-

MISR 형성회로를 이용해 간단히 구성될 수 있기 때문이다. 따라서, 개략적으로 각 TP마다 하나의 2x1 멀티플렉서와 하나의 플립-플롭이 추가된다.

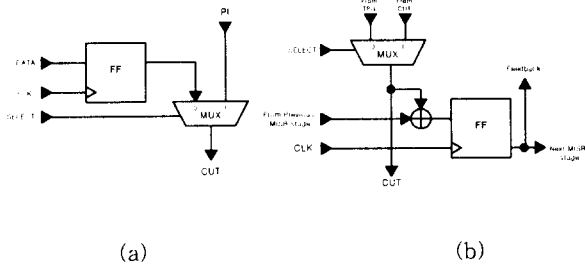


그림 7 PI와 TP에 추가될 부가회로
(a) PI에 추가될 부가회로 (b) TP에 추가될 부가회로
Fig. 7 Additional circuits on PI and TP
(a) Additional circuit on PI (b) Additional circuit on TP

7. 구현 및 실험

본 논문에서 제안한 TP 설정 방법을 알고리즘화하여 표준 C/C++로 구현하였으며 이 프로그램을 'Test Point(TP) Finder'라 명하였다. 'TP Finder'는 제안한 방법과 정리들을 수용하며, 대상회로의 노드 연결정보를 이용하여 필요한 TP를 찾는다. 'TP Finder'를 이용하여 여러 가지 예제회로에 대한 실험을 수행하였으며, 이 결과를 그림 2와 같은 BIST 구성에 사용하였다. 결과 회로는 제안한 방법의 효율성 검증에 의해 시뮬레이션을 실시하였다. 이 과정에서 각 노드에 대하여 고착-1 고장과 고착-0 고장의 검출 대상으로 삼았다.

표 4는 실험대상회로에 대한 정보를 나타내고 있으며, 이 회로들을 실험한 결과를 표 5에 나타내었다. 표 5에서 MLPRP는 PI 그룹화를 수행하지 않았을 경우의 의사-무작위 패턴 테스트에 소요되는 테스트 클럭 수를 의미한다. 이 실험의 수행 환경은 Intel Pentium™ 150Mhz 프로세서를 탑재한 PC이며 수행 시간은 각 3회 실시한 평균 시간이다. 표에서 크게 테스트 시간과 부가 하드웨어 면적의 두 가지 항목을 중심으로 비교하였다. 비교기준은 PI 그룹화하지 않은 MLPRP 테스트 결과이다. 테스트 시간 비교는 테스트시간 감소율(TTRR)을 통해서, 부가하드웨어 면적 비교는 CUT 자체와 PI 그룹화를 수행하지 않은 MLPRP구성시의 면적과 비교하였다. 또한, Rel.H/O로 본 논문에서 제시한 방법에 소요되는 하드웨어 오버헤드가 원회로(Org)또는 MLPRP회로에 대해 어느 정도인가를 %로 표시하였다. 표에서 보듯이 실험 결과는 매우 만족스러우며, 특히 C7552 회로의 경우에는 12.04%의 하드웨어면적의 증가로 $1/2^{151}$ 의 테스트시간 감소효과를 보였다. 대상 회로가 작을 경우에는 하드웨어면적의 증가가 상대적으로 크지만 대상 회로가 클수록 하드웨어 부가율이 감소하며, 테스트시간 감소효과에 비해서 면적증가 문제는 그리 큰 문제가 되지 않을 것으로 생각된다. 결과적으로

제안한 방법은 대형회로의 테스트시간 감소에 매우 효과적인 방법이라 할 수 있다.

한 가지 더 언급할 것은, 5장에서 2진 해독기의 예제에서 두 개의 TP를 통해 테스트 시간을 2^{32} 에서 2^6 으로 줄인 경우를 보였는데, 표 5에서는 네 개의 TP를 사용하여 2^9 으로 시킨 결과를 보이고 있다. 이 예에서 보듯이, 제안한 방법은 요구에 따라 테스트시간 감소와 부가면적 증가 사이에서 절충할 수 있는 여지를 제공한다. 즉, 많은 TP가 선택되면 부가면적은 증가하지만 그에 반해 테스트시간은 더욱 감소시킬 수 있는 것이다. 제작된 'TP Finder' 역시 이러한 요구를 실행 중에 처리해 줄 수 있는데, 사용자가 원하는 #PIG를 입력하여 실행시키는 것이 가능하다.

8. 결 론

본 논문에서는 BIST의 테스트시간을 줄이기 위한 새로운 접근방법을 제안하였다. 이 방법은 기본적으로 PI를 그룹화하여 대상회로를 작은 부분회로로 나누어 콘을 형성시킨다. PI 그룹화는 본 논문에서 정의한 테스트 포인트라는 특정한 노드를 통하여 이루어진다. 제안한 방법은 표준 C/C++언어 기반에서 구현되었으며 실험은 PC상에서 이루어졌다.

여러 대상회로에 대해 실험한 결과, 제안한 방법은 테스트 시간을 최대 $1/2^{151}$ 으로 감소시킬 수 있었으며, 대부분의 회로에서 매우 만족스러운 결과를 얻었다. 본 방법은 테스트시간 감소와 부가 면적 증가 사이에서 절충이 가능하며, 테스트 포인트의 수(부가회로의 양)와 테스트시간 감소간에 상충관계를 가진다. 따라서 원하는 설계 목적에 따라 테스트 포인트의 개수를 정할 수 있다. 또한 대상회로의 크기가 커질수록 하드웨어 부가율이 감소함을 실험적으로 밝힐 수 있었다.

결론적으로, 제안한 BIST 기술은 하드웨어 제작비용의 감소추세와 더불어 VLSI, ULSI와 같은 대규모 회로에서 더욱 효율성이 증가될 것이며, 향후의 테스트 기술 동향에서 유용한 기술로 사용될 수 있을 것으로 예측된다.

표 4 실험대상 회로 정보

Table 4 Information of circuits for experiments

CUT	CUT Information			
	#PIs	#POs	#Nodes	#Comp.
16-bit-Adder	32	16	70	86
8-bit-comparator	19	1	56	57
Binary decoder with enable	9	6	10	16
8-bit ALU	20	4	105	109
4-bit ALU	12	4	111	115
32-bit parity generator	32	1	17	18
C17	5	2	4	6
C432	36	7	153	160
C499	41	32	170	202
C880	60	26	357	383
C1356	41	32	514	546
C1908	33	25	855	880
C3540	50	22	1647	1669
C5315	178	123	2184	2037
C6288	32	32	2384	2416
C7552	207	108	3405	3512

#PIs : number of Primary Inputs, #POs : number of Primary Outputs, #Nodes : number of interior nodes, #Comp. : number of components

감사의 글

본 논문은 1999년도 광운대학교 교내학술연구비에 의하여 연구되었음.

참 고 문 헌

- [1] Gandi Singer, "VTS 97 Keynote: The Future of Test and DFT", IEEE Design & Test, pp. 11-14, July-Sep. 1997.
- [2] Bill Bottoms, "The Third Millenium's Test Dilemma", IEEE Design & Test, pp. 7-11, Oct.-Dec. 1998.
- [3] Vishwardi D. Agrawal, "A D&T Roundtable: Built-In Self Test for Designers", IEEE Design & Test, pp. 113-122, July-Sept. 1997.
- [4] Chih-Ang Chen and Sandeep K. Gupta, "Efficient BIST TPG Design and Test Set Compaction via Input Reduction", IEEE Trans. on CAD of Integrated Circuits and Systems, Vol. 17, No. 8, pp. 692-705, Aug. 1998.
- [5] Robert A. Frohwork, "Signature Analysis: A New Dogital Field Service Method", HP Jour., pp. 2-8, May 1977.
- [6] Vishwardi D. Agrawal, Chares R. Kime, and Kewal K. Saluja, "A Tutorial on Built-In Self Test, Part I: Principles", IEEE Design & Test, pp. 73-82, March 1993.
- [7] Vishwardi D. Agrawal, Chares R. Kime, and Kewal K. Saluja, "A Tutorial on Built-In Self Test, Part II: Applications", IEEE Design & Test, pp. 69-77, June 1993.
- [8] Brend Konemann, Joachim Mucha, and Gunther Zwiehoff, "Built-In Logic Bock Observation Techniques", IEEE Test Conference, pp. 37-41, 1979.
- [9] Brend Konemann, Joachim Mucha, and Gunther Zwiehoff, "Built-In Test for Complex Digital Integrated Circuits", IEEE J. of Solid-State Circuits, Vol. SC-15, No. 3, pp. 315-319, June 1980.
- [10] Li-Ren Huang, et al., "Gauss-Elimination-Based Generation of Multiple Seed-Polynomial Pairs for LFSR", IEEE Trans. on CAD of Integrated Circuits and Systems, Vol. 16, No. 9, pp. 1015-1024, Sept. 1997.
- [11] Chih-Ang Chen and Sandeep K. Gupta, "Efficient BIST TPG Design and Test Set Compaction via Input Reduction", IEEE Trans. CAD of Integrated Circuits and Systems, Vol. 17, No. 8, pp. 692-705, Aug. 1998.
- [12] Jaun-Chih Tsai, et al., "Efficient Test-Point Selection for Scan-Based BIST", IEEE Trans. VLSI Systems, Vol. 6, No. 4, pp. 667-676, Dec. 1998.
- [13] Albecht P. Stroele and Hans-Joachim Wunderlich, "Hardware-Optimal Test Register Insertion", IEEE Trans. on CAD of Integrated Circuits and Systems, Vol. 17, No. 6, pp. 531-539, June 1998.
- [14] W. W. Peterson and E. J. Weldon, Jr., *Error-correcting Codes, 2nd Ed.*, The MIT Press, Cambridge, MA, 1972.
- [15] S. W. Golomb, *Shift Register Sequences*, the Revised Ed., Aegean Park Press, Laguna Hills, CA, 1982.

표 5 실험결과
Table 5 Experimental result

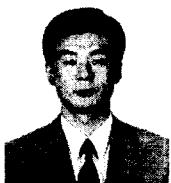
CUT	#of TPs	CPU time(sec)	Test time			H / O		Rel. H/O (%)		Fault Coverage(%)
			MLPRP	this	TTRR(%)	MLPRP	this	Org.	MLPRP	
16-bit-Adder	4	0.71	$\sim 2^{12}$	$\sim 2^{10}$	0.0015	80	72	83.72	-10.00	100.00
8-bit-comparator	8	0.65	$\sim 2^{19}$	$\sim 2^{11}$	0.39	39	47	82.45	20.51	100.00
Binary decoder with enable	2	0.14	$\sim 2^9$	$\sim 2^7$	25.05	24	26	162.50	8.33	100.00
8-bit ALU	12	1.10	$\sim 2^{20}$	$\sim 2^{12}$	0.39	44	60	55.04	36.36	100.00
4-bit ALU	27	1.30	$\sim 2^{12}$	$\sim 2^9$	6.25	28	78	67.82	178.50	100.00
32-bit parity generator	4	0.16	$\sim 2^{12}$	$\sim 2^9$	0.000012	65	50	277.78	-23.08	100.00
C17	2	0.06	$\sim 2^9$	$\sim 2^3$	25.8	12	14	233.33	16.60	100.00
C432	11	1.49	$\sim 2^{30}$	$\sim 2^{18}$	0.00038	79	83	51.90	5.06	100.00
C499	12	2.97	$\sim 2^{11}$	$\sim 2^{10}$	2.29e-6	114	113	55.90	-0.87	100.00
C880	12	5.77	$\sim 2^{31}$	$\sim 2^{31}$	9.09e-11	146	130	33.94	-10.96	100.00
C1356	10	11.05	$\sim 2^{11}$	$\sim 2^{14}$	7.45e-7	114	107	19.59	-6.14	100.00
C1908	10	14.11	$\sim 2^{11}$	$\sim 2^{17}$	0.0015	91	95	10.79	4.39	100.00
C3540	13	73.40	$\sim 2^{31}$	$\sim 2^{18}$	2.32e-8	122	116	6.95	-4.91	100.00
C5315	52	221.40	$\sim 2^{178}$	$\sim 2^{27}$	3.50e-44	479	432	18.72	-9.81	100.00
C6288	44	443.00	$\sim 2^{32}$	$\sim 2^{24}$	0.390	96	176	7.28	83.33	100.00
C7552	26	561.10	$\sim 2^{207}$	$\sim 2^{36}$	3.50e-44	522	423	12.04	-18.96	100.00

#TPs : number of imposed Test Points,

*Rel. H/O : Relative Hardware Overhead(Org. versus this and MLPRP versus this)

※ 위 표에서 입력 그룹의 크기는 식(10) 만을 적용하여 수행한 결과임

저 자 소 개



김 동 옥 (金東郁)

1960년 8월 23일생. 1983~1985년 2월 한양대 전자공학과 졸업(학사, 석사) 1991년 미국 Georgia Institute of Technology 전기공학과 졸업(공학) 1992년~현재 광운대 전자재료공학과 부교수 재직중, Professional activities of SSCS/EDS Joint chapter of IEEE Seoul section chief 활동. <주관심 분야 : VLSI 설계, CAD, Digital Testability 등>

Tel : 02-940-5167, Fax : 02-919-3940

E-mail : vlsicad@daisy.kwangwoon.ac.kr



최 병 구 (崔炳九)

1972년 8월 15일생. 1999년 2월 광운대 제어계측공학과(학사) 현재 광운대 전자재료공학과 석사과정. 주관심 분야 : VLSI설계, CAD, Digital Testability 등

Tel : 02-940-5167, Fax : 02-919-3940

E-mail : cbg1993@chollian.net