論文95-32A-12-27

조합 논리 회로의 경로 지연 고장 검출을 위한 가중화 임의 패턴 테스트 기법

(A Weighted Random Pattern Testing Technique for Path Delay Fault Detection in Combinational Logic Circuits)

許 龍 珉 *, 林 寅 七 *

(Yong-Min Hur and In-Chil Lim)

요 약

본 논문에서는 조합 논리 회로의 경로 지연 고장 검출을 위한 새로운 가중화 임의 패턴 테스트 생성기법을 제안한다.

제안한 경로 지연 고장의 테스트 생성 기법은 대상회로의 기본 논리소자들에 대한 주입력의 신호천이 확률값 계산시, 대상회로의 구조적정보와 의사 임의 테스트 벡터에 의해 생성된 벡터를 각각 초기화 벡터(initialization vector)와 테스트 벡터(test vector)의 생성에 이용한다. 회로내 논리소자간의 레벨차이를 고려하여 가중값을 논리신호선에 할당하므로써 보다 많은 수의 경로를 활성화시키도록 한다. ISCAS '85 벤치마크회로를 사용하여 제안하는 기법이 기존 테스트 방식보다 테스트 길이와 고장 검출면에서 우수함을 보인다. 또한, 최장경로길이(longest path)와 이에 근접한 경로들에 대하여 로보스트(robust) 테스트 벡터를 더 많이 생성함을 보인다.

Abstract

This paper proposes a new weighted random pattern testing technique to detect path delay faults in combinational logic circuits.

When computing the probability of signal transition at primitive logic elements of CUT(Circuit Under Test) by the primary input, the proposed technique uses the information on the structure of CUT for initialization vectors and vectors generated by pseudo random pattern generator for test vectors. We can sensitize many paths by allocating a weight value on signal lines considering the difference of the levels of logic elements. We show that the proposed technique outperforms existing testing method in terms of test length and fault coverage using ISCAS '85 benchmark circuits. We also show that the proposed testing technique generates more robust test vectors for the longest and near-longest paths.

I. 서 론

최근 VLSI의 기술의 발달로 회로의 테스트가 매우 중요한 문제로서 부각되고 있다. 이는 외부에서 접근할 수 있는 입출력핀수는 크게 증가되지 않는 반면에 회로의 복잡도와 크기가 증가하고 있기 때문이며 따라서 회로 자체내에서 테스트를 수행하여 회로 테스트 시간을 줄이고 효율적인 테스트 수행을 위한 BIST (Built-In Self-Test) 와 DFT (Design for Testability)방식에 관한 연구가 활발하다.

* 正會員, 漢陽大學校 電子工學科

(Dept. of Elec. Eng., Hanyang University)

接受日字:1995年9月1日, 수정완료일:1995年11月17日

일반적으로 stuck-at 테스트는 회로의 기능적인 면만을 테스트하므로 회로소자의 지연 및 대상회로의 타이밍에 관련된 성능을 테스트할 수 없다. 반면에 지연 테스트 (delay test)는 테스트 절차와 시간은 증가되지만 회로의 기능적인면 뿐만 아니라 시스템 클럭주기 내에 회로가 정상적으로 출력값을 발생시키는 지를 테스트할 수 있는 고신뢰도의 테스트 방식이라 할 수 있다. 지연고장 테스트는 한 쌍의 테스트 벡터를 필요로 하며 이를 초기화 벡터 (initialization vector), 테스트 벡터 (test vector)라 하고 상승천이와 하강천이를 주입력에서 발생시켜 주출력까지 이 천이가 정해진 시간내에 도달되는지를 검증하게 된다.[1] 또한 지연고장의 고장 모델로는 크게 게이트 지연 고장 (gate delay fault) 과 경로 지연 고장 (path delay fault)으로 나뉘어지며 게이트 지연 고장의 테스트 방식에서는 고장을 일으킬 수 있는 고장크기 (fault size) 가 그리고 경로 지연 고장에서는 테스트 되는 대상 경로 고장수가 주요 관심의 대상이 된다. 경로 지연 고장모델을 대상으로 하는 지연테스트 방식에서는 활성화된 경로상에 존재하는 게이트 지연 고장들을 포함하고 있으므로 보다 높은 신뢰성을 가진 테스트 방식이지만, 회로 크기가 증가하면서 고려되어야 할 경로수가 크게 증가하므로 대상회로의 모든 경로를 테스트 하는데 많은 테스트와 시간이 걸린다. 따라서 적은시간내에 보다 많은 수의 경로들을 신뢰성있게 테스트하는것이 중요한 문제라 할 수 있다. 참고문헌 [2]와 [3]에서는 한 번 지나간 경로들에 대해서 인덱싱을 하여 대상경로들을 제한하는 방식등이 제안되었다. 그러나 고장검출에 오차를 지니는 단점을 가지고 있다.

가중화 임의 패턴 테스트 (weighted pseudo random pattern test)기법은 주로 단일 stuck-at고장 모델에 국한하여 여러 알고리듬들이 제안되어 왔다.[4-14] Waicukauski와 Lindbloom([4] [5]논문의 저자)은 회로 소자의 비제어값(noncontrolling logic value)의 전달확률값을 각 게이트의 팬인수를 고려하여 주출력에서 주입력까지 전파시켜 가중화 값을 구하였으며, [6]에서는 마코프체인(marcov chain)모델을 이용하여 회로내 모든 게이트에서 천이 발생 경우를 고려, 평균한 값으로 주입력에 대한 확률값을 계산하였다. [7]에서 제안된 방식은 임의 패턴 테스트 방식과 결정론적(deterministic) 테스트 알고리듬을 혼용한 방식으로 임의 패턴으로 고장 검출이 어려운 고장(random pattern resistant faults)에 대해서 결정론적인 방법으로 테스트를 구하여 해당되는 주입력선에 확률값을 구하는 방식을 취했다. 또한 [8]논문에서는 비트-플립핑(bit-flipping)방식으로 확률값을 구한다. 최근 [9,10,11] 등의 연구에서는 순수 결정론적인 방법으로만 구한 테스트벡터를 압축(compaction)시켜 확률값을 구하는 방식을 제안하였다. 그러나 이처럼 stuck-at고장과 게이트 지연 고장의 임의 패턴 및 가중화 임의 패턴 테스트 기법에 대해서는 많은 연구가 이루어져왔지만 경로 지연 고장에 대해서는 아직까지 문헌상으로 연구 발표되어 있지 않다. 경로 지연 고장의 경우는 stuck-at고장과 게이트 지연 고장의 경우보다 많은 수의 테스트 벡터가 필요하고 보다 신뢰성있는 로보스트(robust)의 테스트 생성이 중요하며 특히, 최장 경로와 이에 근접한 고장 모델에 대해서 많은 수의 로보스트 테스트 생성이 중요한 문제로서 고려해야 할 사항이다. 따라서 본 논문에서는 이러한 경로 지연 고장의 신뢰성있는 가중화 테스트 생성 방식과 고장시뮬레이션 결과를 고찰해본다. Ⅱ장에서는 기존의 임의 패턴 테스트 벡터 생성방식에 관하여 알아보고 Ⅲ장에서는 경로 지연 고장 검출을 위한 가중화 임의 테스트 생성 절차를 설명하고 Ⅳ장에서는 경계주사구조(IEEE 1149.1)의 적용예, Ⅴ장은 신뢰도 향상 실험예를 보인다. Ⅴ장에서는 결론으로 맺는다

Ⅱ. 임의 패턴과 가중화 임의 패턴 테스트

1. 임의 패턴 테스트 방식

디지탈회로의 테스트방식중의 하나로 대상회로의 구조적정보나 기능적인 정보없이 입력을 인가하고 출력측에서 결과 데이타값을 압축하는 방식으로 수행된다. 이때 인가되는 입력원은 LFSR(Linear Feedback Shift Register) 또는 CAR(Celluar Automata Register)로서 의사임의(pseudo random) 패턴을 만들어내고 출력측에서는 MISR(Multiple Input Signature Register)등으로 응답을 압축하게 된다. 이 임의 패턴 테스트방식의 특징으로는 테스트 데이타를 위한 저장공간을 최소화 할 수 있는 반면에 원하는 고장 검출율을 얻기위해 인가되는 테스트 벡터수의 조정이 힘들고 특히, 높은 고장 검출율을 얻기위해서는 많은 수의 테스트벡터가 인가되야 하는 단점을 가지고

있다. 또한, 임의 패턴원으로서 CAR을 사용할 수 있으며 LFSR 또는 CAR 모두 2^n - 1 의 패턴을 생성할 수 있다. n은 레지스터 수이다.

2. 가중화 임의 패턴 테스트 방식

가중화 임의 패턴 테스트는 순수한 임의 패턴 테스트 방식에 비해 높은 고장 검출율을 보이며 특히 적은 수의 테스트 벡터로서 많은 고장 수를 검출하는 특징을 갖고 있다. 예를 들어 5입력 OR게이트의 stuck-at 고장 검출시 00001,00010,00100,01000,10000, 00000으로서 충분하다. 그러나 입력측에 0.5의 논리 1또는 0값의 확률을 발생시키는 원시다항식(primitive polynomial) LFSR또는 CAR로서 테스트할 경우, 보다 논리 0값을 발생시킬 수 있는 확률을 높여 5입력 OR게이트의 경우 (n-1) / n의 0의 확률로서 인가하게 되면 임의 패턴 테스트보다 적은 수의 테스트벡터로서 많은 고장을 검출할 수 있어 테스트시간을 단축시킬 수 있다.

그림 1의 (a), (b)는 일반적으로 디지탈 회로에서 가장 많이 쓰는 입력패턴원의 그림이다.

(a),(b)모두 2^4 - 1의 서로다른 테스트벡터를 생성할 수 있으며 특히 CAR의 레지스터내의 수치는 CAR를 구성하는 규칙이다. 150,90은 각각 $C_i(t+1) = C_{i-1}(t) \oplus C_i(t) \oplus C_{i-1}(t)$, $C_i(t+1) = C_{i-1}(t) \oplus C_{i+1}(t)$ 로서 양측의 0은 경계조건으로서 2^n - 1의 벡터를 생성가능하게 한다.[15] ⊕는 mod2연산을 의미한다.

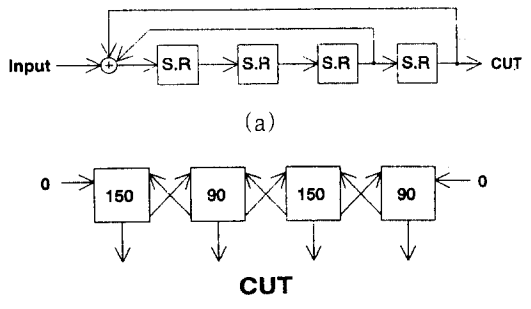

그림 1. LFSR과 CAR 의 블럭선도
(a) LFSR ($X^4 + X + 1$)
(b) 경계조건 0을 갖는 혼성 CAR

Fig. 1. The block diagram of LFSR and CAR.
(a) LFSR ($X^4 + X + 1$)
(b) Hybrid CAR with 0 boundary condition

Ⅲ. 경로 지연 고장 검출을 위한 가중화 임의 패턴 테스트 생성

경로지연 고장 검출을 위한 가중화 임의 테스트 생성은 지연고장 테스트시 인가되는 초기화벡터(V1)와 테스트벡터(V2)의 한쌍의 확률값을 계산한다. 즉, 상승 또는 하강천이를 주입력에 보다 많이 발생시킬 수 있도록 하는 논리 0 또는 1값에 대한 확률수치를 계산해야만 한다. 앞장에서 전술한 바와 같이 가중화 임의 패턴 생성방법들은 크게 두개의 부류로 나뉘어 질 수 있는데, 회로구조 및 각 기본적인 논리소자의 함수를 이용한 방식 그리고 결정론적 테스트 패턴의 조합으로 가중화 값을 구하는 방식이 있다. 기존의 연구에서는 주로 stuck-at테스트와 천이고장(transition fault)에 관하여 연구되어 왔으며 이러한 고장들에 대해서는 두 개의 분류중 후자의 방식이 보다 적은 수의 테스트 패턴으로 많은 고장을 검출할 수 있을 것으로 기대된다. 이는 대상고장의 수가 대상회로의 크기와 선형적으로 증가하기 때문에 결정론적인 방식으로 테스트벡터를 생성하여 생성된 테스트벡터의 1(0)과 0(1), 1(0)과 1(0), 0 또는1 과 X 의 상충 및 중복여부를 따져 간단히 가중화 확률값을 구할 수 있는 것이다. 그러나 경로 지연 고장의 경우 대상회로에 따라 활성화될 수 있는 경로수가 크게 증가하므로 일일이 고장 검출가능한 경로에 대해 테스트 벡터를 구하여 이를 압축할 수 있는지를 따지는 것은 매우 많은 결정론적 테스트 생성시간과 압축 그리고 시뮬레이션 시간이 소요된다. 따라서 간단히 회로의 구조적 정보와 논리소자의 함수를 이용하여 가중화값을 구하는 것이 테스트생성 시간을 줄이는 타당한 접근이라 하겠다. 본 논문에서는 Waicukauski가 제안했던 개념과 유사하게 각 게이트의 비제어값(non-controlling value)과 제어값(controlling value)의 확률값을 계산하여 주출력에서 주입력으로 유도하고 특히 경로 지연 고장의 검출확률을 높이기 위한 방법으로 팬아웃지점에서 발생하는 AND논리소자와 OR논리소자계통의 효과적인 확률값계산 그리고 보다 긴 경로들의 로보스트 테스트 생성증가를 위한 회로내 논리소자 레벨차이에 따른 가중값의 할당등에 관하여 연구한다.

1. 초기화벡터 및 테스트벡터의 가중화 확률값 계산

지연테스트의 초기화벡터와 테스트벡터를 구하기위

해 우선 초기화벡터는 대상회로내의 각 논리소자의 비제어값의 확률을 구한다. 이것은 AND게이트의 경우 입력선에 1값을 모두 인가하게 되면 출력값은 1이 된다. 이때 다음에 오는 입력논리값으로 하여금 출력에서의 0또는 1의 값이 그대로 전파가능하다. 따라서 초기화 벡터는 각 게이트의 비제어값의 확률값으로 정하고 테스트벡터는 이렇게 비제어값으로 할당된 논리값을 천이시킬 수 있어야 한다. 일반적으로 의사임의 패턴 생성기로서 원시다항식 LFSR또는 CAR이 사용되고 이들을 통해 생성되는 0또는 1의 논리값의 확률값은 0.5로 가정한다. 따라서 이들 소스(source)레지스터의 논리값의 생성횟수를 변경시키는 논리를 첨가하여 0과 1사이의 확률값을 조정하게 된다.

$$P_i=\{P_1, P_2, \ldots, P_n\}, \quad T_i=\frac{\sum_{j=1}^{2^n-1}\sum_{i=1}^{n} P_i}{n\times 2^n-1} \cong 0.5, \ (1 \le i \le n) \tag{1}$$

$$WP_i=\{WP_1, WP_2, \ldots, WP_n\}, \ 0 \le I_i=\frac{\sum_{j=1}^{2^n-1}\sum_{i=1}^{n} wp_i}{n\times 2^n-1} \le 1, \ (1 \le i \le n) \tag{2}$$

식 (1)의 T_i 원시다항식 의사 임의 패턴생성기를 통해나오는 논리 1값의 확률값이며 (n 은 레지스터 수) 식 (2)의 I_i는 가중화논리를 거쳐 나오는 확률값이다. 이식에서 P_i는 i번 주입력에 논리0 또는 1값을 가질 수 있는데 n개의 레지스터에서 1이 나올 경우만을 고려한다면 2^n-1 개의 패턴을 인가하였을 경우의 확률은 식(1)과 같다. 식(2)는 가중화논리를 적용하여 주입력에 인가되므로 0과 1사이의 값을 취하게 된다. wp_i 는 LFSR의 P_i가 아닌 확률값을 부여할 수 있는 회로를 통해 나오는 값이다.

표 1에서 제시된 바와 같이 벤치마크에서 사용되는 기본적인 소자 8개와 팬아웃지점에서의 가중값을 계산하였다. N_i(0 또는 1)는 i레벨 게이트의 제어 및 비제어논리값의 확률값이고 N_i+1(0 또는 1)는 i+1 레벨 게이트의 제어 및 비제어논리값의 확률값이다. 또한, α (M_{i+1}/M_i) (식 (3))는 각 논리소자가 주입력에 의존하는 파라메타로서, 해당논리소자에 영향을 줄 수 있는 구조적인 주입력 갯수의 비율이 된다. 따라서 일반적으로 회로의 레벨이 증가할수록 M_i의 값은 커지게 된다. 결국 해당게이트의 제어 및 비제어논리값의 확률값을 주는 주요한 인자는 M_{i+1}/M_i로서 작용하게 된다.

그림 2에서 보는 바와 같이 OR게이트의 주입력선 수는 이전단의 AND게이트의 M_i에 대해서 다음식이 성립한다.

$$\alpha = \frac{M_{i+1}}{M_i} \ge 1 \tag{3}$$

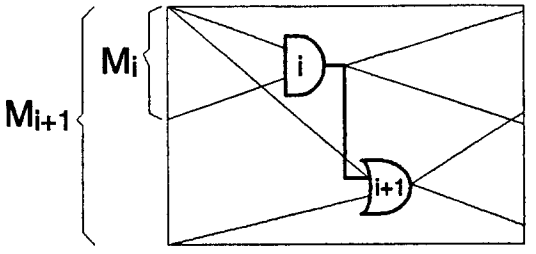

그림 2. 각 게이트에 대한 주입력선의 구조적 관계
Fig. 2. Structual relation of PI signals of each gate.

표 1. 초기화 및 테스트 벡터의 주입력확률 계산규칙
Table 1. Computation rules of prob. of initialization vector and test vector for primary inputs.

gate type and fanout	prob. of initialization vector for		prob. of test vector
	controlling value	non-controlling value	
AND	$N_i(0) = N_{i+1}(0)$	$N_i(1) = N_{i-1}(1) \times \alpha$	P_i
OR	$N_i(1) = N_{i+1}(1)$	$N_i(0) = N_{i-1}(0) \times \alpha$	P_i
NAND	$N_i(0) = N_{i-1}(1)$	$N_i(1) = N_{i-1}(0) \times \alpha$	P_i
NOR	$N_i(1) = N_{i+1}(0)$	$N_i(0) = N_{i-1}(1) \times \alpha$	P_i
XOR	$N_i(0) = N_{i-1}(0) \times \alpha$, $N_i(1) = N_{i+1}(1) \times \alpha$		P_i
XNOR	$N_i(0) = N_{i-1}(1) \times \alpha$, $N_i(1) = N_{i+1}(0) \times \alpha$		P_i
BUF	$N_i(0) = N_{i-1}(0)$, $N_i(1) = N_{i-1}(1)$		P_i
INV	$N_i(0) = N_{i-1}(1)$, $N_i(1) = N_{i-1}(0)$		P_i
FANOUT STEM	$N_i(0) = \sum_{j=1}^{n} N_{i+1}^j(0)$, $N_i(1) = \sum_{j=1}^{n} N_{i+1}^j(1)$		P_i

주입력에 영향을 많이 받는 게이트가 많은수의 경로를 가지므로 회로의 구조적정보를 가중화 확률계산에 도입할 수 있다. 물론, 이러한 수치는 주입력에서 해당 게이트의 i 번 node 까지의 영역이므로 이로부터 주출력까지의 팬아웃수에 따라 회로에 대한 영향력이 달라질 수 있지만 이는 벤치마크회로의 실험으로 대체적으로 해당게이트까지의 주입력에 대한 영향력이 크게 작

용한다는 것을 실험을 통해 알 수 있었다. 그러므로 주출력에서 제어값와 비제어값의 논리값을 각각 (1,1)로 세트시키고 이값을 주입력쪽으로 전파시켜 나가면 CUT(Circuit Under Test)의 주입력에 각 제어값과 비제어값을 생성할 수 있다.

예를 들어 3입력 AND게이트의 회로만을 고려한다면 M_{i+1}/M_i 는 3/1로 각 입력 신호선에 제어값과 비제어값이 각각 (1,3)이 되어 3/4의 논리 1의 확률값이 입력에 가중화될 수 있다.

이때 M_i의 계산은 미리 회로내의 모든 게이트에서 계산해야하는데 이계산에 드는 게이트수가 증가할수록 이를 계산하는 시간이 팬아웃과 재수렴(reconvergent)게이트의 존재로인해 한 개의 게이트를 통해 여러번 계산되므로, 지수함수적으로 크게 증가하므로 휴리스틱한 방법으로 주입력에서 주출력으로 선형적으로 증가시키는 알고리듬으로 이 전처리시간을 줄인다. 알고리듬 Pro_pre_back 는 계산량이 n의 크기를 갖는다. 주입력 노드를 시작으로 각 게이트의 Mi값은 입력 되는 신호선의 수로만 계산되어 진다. 즉 재수렴되는 구조에 상관 없이 현재 게이트의 Mi값은 현재 게이트를 입력으로 하는 그 이전단의 Mi값의 합으로 계산되어 진다. 아래 알고리듬에서 i=i->next는 레벨화된 linked 리스트 구조에서의 다음에 연결된 주입력노드 또는 게이트를 의미한다.

```
알고리듬 Pro_pre_back
i = 1st PI node /* i is a levelized primitive gate */
while( i != Null )
{
  if( i == PI_node ) {
                    i -> c = 1;
                    i = i -> next ;
                        continue ;
                    }
  else {
       i_input = i -> input ; /* i_input is the input
                                 node for current node
                                 */
       while( i_input != Null )
            { i -> c = i -> c + i_input -> c ; }
              i_input = i_input -> next ;
    }
      i = i -> next ;
}
```

2. 경로 지연 고장의 로보스트 테스트와 넌로보스트 테스트(nonrobust test)

경로 지연 고장 테스트 생성과 고장 시뮬레이션시 가능한 많은 로보스트 테스트를 생성하고 검출하여야만 하는데 이는 경로 지연 고장 테스트의 정의가 경로상에서 발생할 수 있는 모든 지연고장의 결함(defect)을 모두 검출할 수 있기 때문이다. 따라서 가중화 임의 패턴 테스트 생성시에도 넌로보스트 테스트 보다 로보스트 테스트 생성에 중점을 둬야 한다.

실험결과에서도 순수하게 LFSR 또는 CAR로만 테스트벡터를 인가하였을 경우 넌로보스트의 테스트벡터 증가에는 큰 기여를 하지만, 로보스트 테스트 생성에는 만족할 만한 결과를 얻지 못함을 알 수 있다. 따라서 가중화 임의 패턴 테스트 생성시 초기화 벡터로서 테스트 벡터를 동일한 확률값을 인가하게 되면 전체회로의 활성화에 큰 기여를 하지 못한다. 또한 주입력에 인가되는 초기화 벡터의 반대확률(예를들어 V1은 0.9 V2는 0.1)을 인가하더라도 이러한 패턴양상이 전체 주입력에 반복됨으로 좋은결과를 얻을 수 없었다. 이는 초기화벡터가 각 주입력선에 비제어값의 논리값을 인가하였으므로 그 다음에 오는 테스트벡터에 의해 천이 지연의 기회가 높아지며 또한 회로의 깊이(depth)가 깊어질수록 천이를 일으킬 수 있는 주입력선의 천이신호선의 상대적인 위치도 증가하므로 주입력선 전체에 고른 상승 및 하강천이의 기회를 주어야만 한다.

1) 팬아웃 지점에서의 가중값 계산

그림 3에서와 같이 A지점에서의 0과 1의 가중값계산을 위해 waicukauski가 제안한 이론에 의하면 i+1 레벨의 0과 1의 가중값중 큰값을 선택하여 주 입력으로 전파시킨다. 반면 제안된 방식에 의해서는 모두 팬아웃가지(fanout branch)에 전파되어온 가중값을 더하여 계산하므로(표 1참조) 어느 한 가지에 치우치지 않는 천이기회를 제공하여 준다. 그림 3의 팬아웃스템(fanout stem) A에서 제어 및 비제어값은 (5,7)이므로 이는 결국 7/12의 1의 확률값을 A지점의 가중화값으로 고려하게 되어 B와 C신호선에 보다 정확한 가중값을 제공하여 준다. 그러나 가장 큰 값으로 팬아웃지점에 값을 할당한 경우 $(4, 4)^{\#}$ 의 4/8 확률값을 계산함으로 보다 세분화하여 차별화할 수 있는 기회를 상실하게 되는 것이다. 이는 V장에서 여러 ISCAS 85의 실험결과의 예에서도 보다 향상된 고장 검출수를 갖게된다.

이는 팬아웃가지의 여러값들중 어느 한개의 값이 크면 다른 가지의 상대적인 값차이는 무시하고 하나의 값만을 선택하여 주입력으로 전파시킴으로 인한 부정

확한 확률값의 발생을 지양할 수 있으므로 벤치마크 회로의 실험결과에서도 보다 향상된 결과를 얻을 수 있었다.

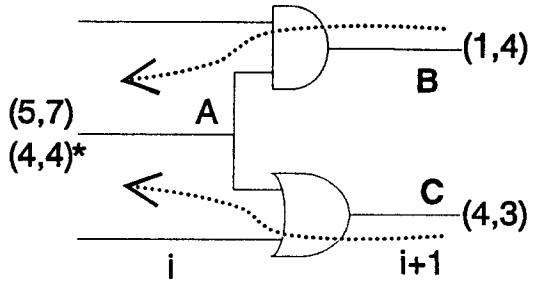

그림 3. 팬아웃지점 A의 가중값 계산
Fig. 3. Weight value computation of fanout stem A.

2) 회로내 논리소자의 레벨차이에 따른 가중값 부여

그림 4와 같은 회로에서 게이트 A와 B는 1레벨의 차이가 있으며 결국 B 게이트의 출력값은 A 게이트의 출력과 3번 주입력선에 의해 결정되어진다. 단순히 주출력에서 표 1에 의해 의해서만 계산을 주입력까지 수행하면 동일한 (1,3)의 값을 얻으므로 결국 논리 1값에 대한 확률값 3/4을 인가하게 되어 주입력 1,2,3 모두에 동일한 가중값을 부여하게 된다.

경로 지연 고장의 경우 주입력선 1,2 -> A -> B -> PO의 경로가 주입력선 3 -> B -> PO의 경로보다 길게 되므로 B번 게이트에서 주입력선으로 가중값을 전파시 A번 게이트보다 높은 가중값을 할당하여 전파시키면 상대적으로 레벨차이가 적은 경로들의 활성화가 높아지게 된다. 따라서 보다 긴 경로들과 로보스트 테스트 생성이 용이하여진다.

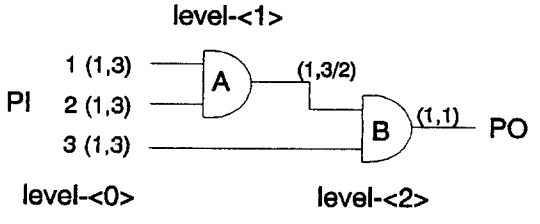

그림 4. 레벨차이에 따른 가중값 할당
Fig. 4. Weight assignment according to difference of level.

본 논문에서는 위의 예와 같이 레벨차이가 큰 지점(fanout 또는 fanout free region)에 원래의 계산규칙에 의거한 값할당 보다 더 큰 가중값을 할당하여 실험한 결과 향상된 고장 검출율을 얻을 수 있었다. (표 3 참조)

3. 가중화 임의 패턴 테스트 시스템

다음의 일련의 절차는 제안된 WAVE 시스템 (Weighted rAndom Vector gEnerator system)의 테스트 생성 및 시뮬레이션의 처리과정이다.

1) 대상회로의 네트리스트(netlist) 를 입력
2) 전처리과정
   Mi+1 / Mi 결정 : PI -> PO
   신호선의 가중화값 결정 : PO -> PI
3) 가중화 패턴 생성기 결정 : LFSR 또는 CAR
4) LFSR 또는 CAR의 레지스터수와 테스트패턴수 및 입력 가중값 (3, 5, 7 논리값중) 결정
5) 테스트 생성 :
   V1 계산된 벡터
   V2 의사 임의 패턴 레지스터로 부터 발생된 벡터
6) 경로지연 고장 시뮬레이션 수행
7) 검출된 로보스트와 넌로보스트 경로 지연 고장수 확인
8) 패턴이 모두 인가되었는가 ? (예 : (9), 아니오 : (5))
9) 종료

전처리 과정시 수행되는 (2)의 과정은 주입력에서 주출력으로 계산되어지고, 신호선에 더해지는 가중화값은 주출력에서 주입력으로 계산되어지는데 계산순서는 각 게이트의 단위 지연과 팬아웃 가지에 선형적으로 비례하는 팬아웃스템의 지연을 고려, 주입력에서 주출력으로 레벨화된 회로의 논리소자의 순서로 계산된다. (4)의 결정시 의사 임의 벡터 발생기로서 레지스터수에 따른 원시다항식 함수로 구성하고 이를 사용자가 레지스터수를 직접 선택할 수 있고 또한, 입력 벡터의 확률에 따른 가중치로서 3,5,7의 가중된 입력값에서 한 개의 값을 선택할 수 있도록 하였다. 이때 사용되는 가중값으로는 다음과 같다.

3 가중값(weight value) =
{ 0.15, 0.5, 0.85 }
5 가중값(weight value) =
{ 0.03, 0.25, 0.5, 0.75, 0.97 },

7 가중값 (weight value) =

{ 0.125, 0.25, 0.5, 0.625, 0.75, 0.875 }

가중값이 증가할수록 이에따른 하드웨어가 증가하며, 가중값이 감소하면 상대적으로 고장검출수는 작아지고 가중화 하드웨어는 감소된다. 예를들어 3 가중값을 선택하였다면 주출력에서 주입력으로 계산되어온 신호선의 확률값이 0.05이였다면 대표값으로 0.15를 선택하도록 한다.

Ⅳ. 가중화 임의 패턴 발생기

전장에서 언급한 바와 같이 가중화 임의 패턴 생성기는 일반적으로 LFSR, CAR로 구성할 수 있는데 이를 레지스터원으로 부터 생성되는 벡터값에 가중화 논리를 추가시켜 이로 인해 생성되는 논리값을 CUT에 인가한다. 대상회로에 인가되는 확률계산 하드웨어는 다음의 공식들로서 용이하게 이뤄진다. 즉 AND게이트의 경우와 OR게이트의 경우는 식(4)로 원하는 확률값의 생성논리를 구성할 수 있다. 그림 5는 IEEE 1149.1의 경계주사 아키텍처의 레지스터 셀에 적용될 수 있으며 BIST논리로서 역시 적용가능하다. Pi 는 주입력선 i의 확률값이다.

$$\text{AND} : \prod_{i=1}^{k} P_i \ , \quad \text{OR} : 1-\prod_{i=1}^{k}(1-P_i) \qquad (4)$$

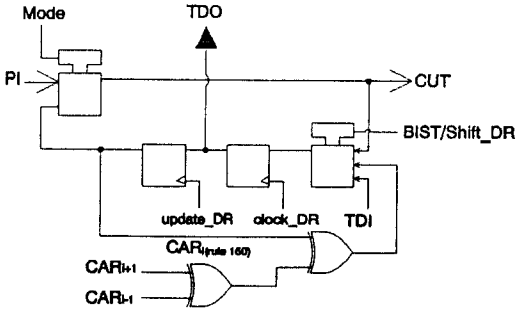

그림 5. CAR 기능을 갖는 입력 경계 주사 셀의 논리도

Fig. 5. Logic diagram of the input boundary-scan cell with CAR function.

그림 6에서 볼 수 있듯이 가중화된 논리를 시스템논리 입력이전에 추가할 수 있으며 (b)의 그림에서는 AND게이트를 통과해서 나오는 논리1에 대한 확률값은 0.25가 된다. 이렇게 미리 전처리과정을 거쳐 각 주입력에 대한 확률값을 구하면 간단한 논리게이트를 첨부시켜 원하는 확률값을 얻게된다. 이때 입력 가중값의 구간을 증가시킬수록 정밀한 확률값은 얻을수 있지만 부가하드웨어는 증가하고 이로인하여 지연시간이 증가하므로 설계시 이를 상반관계(trade-off)로서 신중하게 고려해야 할 파라메타가 된다.

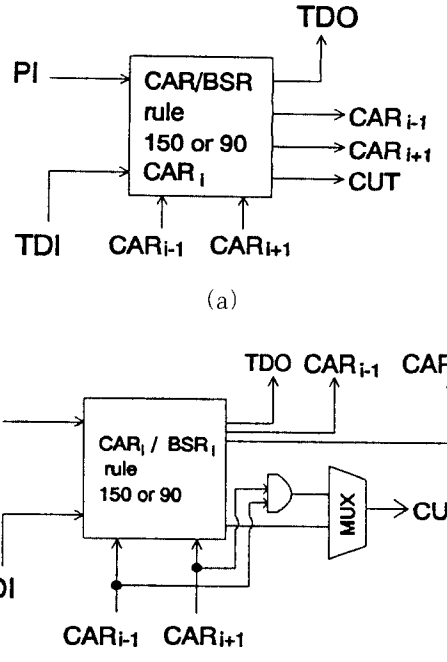

그림 6. CAR과 BSR 셀의 블럭선도
(a) 비가중화된 CAR/BSR 의 블럭선도
(b) 0.25로 가중화된 CAR/BSR의 블럭선도

Fig. 6. Block diagram of a CAR/BSR cell.
(a) Bock diagram of a non-weighted CAR/BSR
(b) Block diagram of a 0.25 weighted CAR/BSR

Ⅴ. 실험 및 결과

본 절에서는 ISCAS85 벤치마크 회로[16]를 대상으로 경로 지연 고장검출을 위해 생성된 테스트 패턴을 고장 시뮬레이션한다. 경로 지연 고장 시뮬레이션으로는 병렬 패턴 시뮬레이션 방식에 기본하여 S0, S1,

R0, R1, U0, U1의 신호논리값을 사용하여 로보스트 테스트시 사용하였으며, 넌로보스트의 경우에는 축약된 N0,N1,U0,U1의 값을 사용한다. 즉, S0,S1은 테스트시 일정하게 0또는 1을 유지하는 신호값이며 R0,R1은 하강천이 또는 상승천이 신호를 로보스트하게 전파하는 신호선에 그리고 U0(U1)은 최종값이 0(1)이고, S0(S1)또는 R0(R1)으로 할당될 수 없는 신호를 표현한다. NO(N1) 은 넌로보스트하게 전파되는 하강천이(상승천이)에 , U0(U1)은 테스트할 수 없는 경로중에서 최종값이 0(1)인 신호선에 각각 할당된다. 실험에 사용된 입력 벡터는 인가된 벡터수가 10,000개 일 경우 9999쌍의 초기화벡터와 테스트벡터가 인가되도록 하였다. 즉, 두번째 인가되는 벡터는 테스트 벡터이면서 다음 경로 지연 고장을 검출하는 초기화 벡터로 사용된다. 시뮬레이션 환경으로 썬스파크2(Sun Sparc-2(24M))를 사용하였으며, 표 7과 8의 실험에서는 썬스파크20(Sun Sparc-20(32M))를 사용하였다.

실험은 다음의 5가지 측면을 고려하여 수행하였다.

1) 제안된 시스템의 성능평가와 로보스트 고장 검출수를 증가시키기 위한 조건을 부여한 경우 (표 3)
2) 입력 테스트 벡터의 발생회로를 다양화 하였을 경우 (표 4, 5)
3) 서로다른 입력 가중값에 의한 실험 (표 6)
4) 동일한 고장 검출율을 얻기위해 의사임의패턴수와 가중화된 테스트 패턴수와의 비교 (표 7)
5) 경로길이에 따른 검출된 고장 분포수 (표 8)

표 2는 사용된 벤치마크회로의 사양을 표시한 것으로 Gate, PI, PO는 각각 회로게이트수, 주입력, 주출력 신호선의 수를 표시한다.

실험 1) 제안된 시스템의 시뮬레이션 성능 비교

표 3은 본논문에서 제안된 WAVE시스템과 [4]논문을 적용하여 생성된 10,000개의 테스트 패턴을 각각 사용하여 고장 시뮬레이션을 수행하였을때 구한 로보스트/넌로보스트 고장 검출 수와 수행시간의 비교표이다. 표에서 보드시 제안된 WAVE시스템이 보다 많은 수의 고장을 검출한 것을 알 수 있으며, 수행시간 또한 크게 감소한 것을 알 수 있다. 표의 여섯번째 열의 WAVE+의 실험치는 전장에서 언급한 회로내 게이트간의 평균 레벨 차이를 고려하여 시뮬레이션한 것으로 로보스트의 고장 검출수가 WAVE시스템의 경우보다 증가한 것을 알 수 있다. ALF는 팬아웃을 갖는 모든 게이트들 중에서 팬아웃스템(stem) 출력을 가진 게이트와 팬아웃가지(branch)를 입력으로 하는 게이트들간의 평균 레벨 차이를 의미하고, ALNF는 팬아웃이 없는 영역의 게이트들간의 평균 레벨 차이의 값을 나타내는데 c3540의 회로를 제외한 모든 회로에서 ALF값이 ALNF값보다 크다는 것을 알 수 있다.(예를 들어 c1355의 경우 ALF는 7.07, ALNF는 3.0) 그러므로 전장에서의 표 1 계산시 팬아웃지점을 만났을때 레벨차이가 2이상 차이가 나면 더해져야 할 가중값에 50%의 값을 더 가중하여 전파시키고, c3540의 경우에는 반대로 팬아웃이 없는 영역의 게이트들에서 역시 레벨차이가 2이상이면 가중값을 더 가중하여 전파시켜 주입력에서의 최종 확률값을 구하도록 하는 휴리스틱 조건을 부여한 실험 결과치이다. 이와같은 실험결과는 로보스트고장 검출 수를 증가시키기 위한 시도이며, WAVE+는 보다 향상된 고장 검출율을 얻을 수 있다.

표 2. ISCAS85의 회로특성
Table 2. ISCAS85 benchmark circuit.

CUT	Gate	PI	PO
c432	153	36	7
c499	170	41	32
c880	357	60	26
c1355	546	41	32
c1908	880	33	25
c2670	1193	233	140
c3540	1647	50	22
c5315	2184	178	123
c7552	3513	207	108

실험 2) LFSR과 CAR패턴 생성기를 이용한 시뮬레이션

표 4는 의사 임의 패턴 테스트발생기로서 LFSR과 CAR로 시뮬레이션 했을때의 실험 결과로 LFSR과 CAR은 가중논리를 사용하지않고 V1과 V2의 논리1값의 확률을 0.5로 가정하여 실험하였다. 실험에 사용된 LFSR과 CAR은 벤치마크회로의 주입력선수가 최소 32를 초과하므로 32개의 레지스터를 갖는 원시다항식 함수를 구성하여 회로에 인가하도록 하였다.

표 3. 제안된 WAVE시스템의 고장 시뮬레이션 결과

Table 3. Fault simulation results of the proposed WAVE system.

CUT	[4] robust/ nonrobust	total (sec)	WAVE robust/ nonrobust	total (sec)	WAVE+ robust/ nonrobust	ALF	ALNF
c432	295/6868	18.32	922/7759	12.37	930/7659	5.34	5.0
c499	853/75567	25.2	853/75567	24.58	853/75567	6.72	3
c880	984/6983	2527	912/6324	23.7	1107/5915	7.14	6.39
c1355	50/291604	489.75	50/291604	79.89	266/215492	7.07	3.0
c1908	1016/28594	246.3	1120/93948	53.21	1208/80256	12.09	8.88
c2670	1285/52003	193.3	2185/70867	124.5	2261/70179	4.33	3.01
c3540	2141/259278	4179.19	1660/208663	210.47	1743/192286	5.31	5.61
c5315	4407/185077	277.4	5083/236169	220.88	5415/236972	7.55	6.6
c7552	2562/132832	422.07	3315/132241	218.72	3735/132062	5.43	4.16

(테스트 벡터 수 : 10,000 개)

표 4. 비가중 LFSR,CAR의 의사임의패턴 발생기에 의한 고장 시뮬레이션 결과

Table 4. Fault simulation results with non-weighted LFSR and CAR pattern generators.

CUT	pseudo random(LFSR)		pseudo random(CAR)	
	robust	nonrobust	robust	nonrobust
c432	302	8326	352	9304
c499	30	113879	22	121381
c880	440	6028	515	6803
c1355	28	304184	16	323705
c1908	625	37545	693	39172
c2670	1186	44989	1164	46671
c3540	567	294947	811	352518
c5315	2988	149560	3073	151435
c7552	2105	14209	2127	151414

(테스트 벡터 수 : 10,000 개)

표에서 알수 있듯이 CAR을 이용한 실험결과가 약간 우수하게 나타났다. 이는 기존의 논문에서 stuck-at을 대상으로 한 실험에서와 유사하게 생성된 벡터간의 상관관계(correation)로 인한 결과인 것으로 생각된다. 표 5는 가중값을 부여하였을 경우와(WLFSR,WCAR) 실험1)에서 적용한 레벨차이에 따른 가중값을 50%로 한 경우(WLFSR+,WCAR+)의 실험 결과로서 표 4의 결과보다 로보스트 경로 지연 고장의 수가 크게 향상된 것을 알 수 있다.

실험 3) 다양한 입력 가중값(weight value)에 대한 시뮬레이션

표 6은 Ⅲ장 3의 입력가중값 모델을 기초로 실험하였다. 대체적으로 입력가중값이 증가할수록 좋은 결과를 얻을 수 있으며 가중값을 세분화하여 BIST회로를 설계하는것이 효율적인것으로 설명된다. 그러나 입력가중값이 증가하면 이에따른 하드웨어 오버헤드도 고려해야 하므로 대상회로의 설계자 및 사용자(customer)는 이를 상반관계의 지표로 고려해야 할 것이다.

표 5. 가중화된 LFSR,CAR의 고장 시뮬레이션 결과

Table 5. Fault simulation results by weighted LFSR and CAR pattern generators.

CUT	WLFSR		WCAR		WLFSR+		WCAR+	
	robust	nonrobust	robust	nonrobust	robust	nonrobust	robust	nonrobust
c432	922	7759	983	8099	930	7659	968	7999
c499	853	75567	882	76447	853	75567	882	76447
c880	912	6324	1038	7183	1107	5915	1225	6285
c1355	50	291604	38	288808	266	215492	280	225368
c1908	1120	93948	1080	94958	1208	80256	1106	79707
c2670	2185	70867	2179	71621	2261	70179	2311	69809
c3540	1660	208662	1777	256755	1743	192286	1878	251678
c5315	5083	236169	5151	237115	5415	236972	5513	237081
c7552	3315	132241	3260	13290	3735	132062	3754	132931

(테스트 벡터 수 : 10,000 개)

실험 4) 의사 임의 패턴수와 가중화 임의 패턴 수와의 관계

표 7은 가중논리를 첨가하지 않고 의사 임의 패턴기로만 사용하였을 경우와 표6의 5-value 가중화 논리를 부여하여 시뮬레이션한 경우를 비교해서 얼마만큼의 패턴을 인가하여야만 제안된 시스템의 로보스트고장 검출수를 얻을 수 있는지를 나타낸다. 예를들어 c432의 경우 최소 310,000 이상의 패턴을 인가하여야만 표 6의 922개의 로보스트 고장수를 검출할 수 있다. 그리고 >50이라고 한 것은 실험을 50만개만 주고 행하였으므로 그 이상 많은 수의 패턴을 인가해야한다

는 것을 의미한다. 검출된 고장형태가 주로 짧은 경로들만을 대상으로 검출하는 경우라면 회로자체의 짧은 경로(short path)나 최장경로미만의 길이를 가진 경로의 타이밍 검증에는 유효할 지 몰라도 시스템 클럭 주기에 영향을 주는 최장경로의 타이밍 검증에는 효과적이지 못하다.

표 6. 서로다른 입력 가중값에 따른 시뮬레이션 결과

Table 6. Simulation Result of different input weighted values.

CUT	3 - value		5 - value		7 - value	
	robust	non robust	robust	non robust	robust	non robust
c432	236	6586	922	7759	803	7546
c499	19	70573	853	75567	853	75567
c880	732	5717	912	6324	1009	6160
c1355	492	214901	50	291604	20	395328
c1908	950	66843	1120	93948	1142	84169
c2670	1982	63001	2185	70867	2432	693198
c3540	1572	25172	1660	208663	427	314014
c5315	4458	219866	5083	236169	6435	217760
c7552	1858	133916	3315	132241	5226	138160

(테스트 벡터 수 : 10,000 개)

표 7. 표 6의 4번째 열의 로보스트 값을 얻기위해 인가되야 하는 순수 LFSR패턴 수

Table 7. Pattern number applied to achieve the equivalent 4th coloumn values in table 6.

CUT	c432	c499	c880	c1355	c1908	c2670	c3540	c5315	c7552
No. of pattern	≅31	>50	>50	>50	≅4	>50	>50	≅21	≅11

(단위 : 10,000)

실험 5) 경로길이에 따른 고장 검출 분포

본논문에서는 제안된 경로지연 고장 테스트기법의 경로길이에 따른 고장검출수와의 분포도를 알아보기위해 회로내 가장 긴 경로를 가진 부류를 1.0으로 하고 이 최장경로길이를 10등분하여 5-value 가중값을 갖는 경로 지연 고장 시뮬레이션을 수행하였다. 표 8은 대상회로의 경로길이 크기에 따른 로보스트, 넌로보스트 고장의 검출 분포를 나타내고 있다. 50만개의 패턴을 인가하였으때 0.8이상의 길이를 갖는 경로에서 의사임의 패턴으로는 9개의 로보스트고장을 검출한 반면, 5 가중값을 사용한 방법으로는 388개의 로보스트고장을 검출하였다. 이 표로서 전체적으로 1.0의 경로 길이에 가까운 로보스트 경로 지연 고장을 많이 검출하였다는 것을 알 수 있었다.

앞서 실험한 표들로부터 비가중 의사 임의 테스트에서 넌로보스트의 고장 검출수가 상대적으로 많은 것은 한 쌍의 패턴을 생성하는데 보다 다양한 형태의 패턴쌍을 만들수 있기 때문이며, 반면에 가중화 논리를 부여하여 시뮬레이션 한 경우는 로보스트 고장 검출 증가를 위해 입력측의 어느 한 쪽이 보다 안정(stable)한 값으로 유지시켜야 하는 관계로 테스트 패턴 생성의 기회가 적어지는데 연유한다.

표 8. c880회로의 경로길이에 따른 고장 검출 분포

Table 8. Distribution of detectd faults according to path length for c880 circuit.

path length	pseudo random (LFSR)		WLFSR(5-value)	
	robust	nonrobust	robust	nonrobust
1.0	0	1	0	2
0.9	0	903	105	735
0.8	9	2563	388	2313
0.7	54	5821	1201	5490
0.6	154	8596	1804	8318
0.5	415	10240	2245	9903
0.4	626	11224	2485	10819
0.3	719	11475	2577	11051
0.2	856	11658	2693	11208
0.1	904	11720	2755	11277

(테스트 벡터 수 : 500,000 개)

VI. 결 론

본 논문에서는 조합논리회로의 경로 지연 고장 검출을 위해 대상회로의 천이를 일으킬 수 있는 확률을 향상시키고 하드웨어 구현을 용이하게 설계하기 위한 가중화 임의 패턴 테스트 생성기법을 제안하였다. 제안된 테스트기법은 초기화벡터로서 제어값과 팬인수를 고려하여 주출력에서 주입력까지의 전파확률을 계산하였으며 테스트 백터로서 원시다항식 함수 발생기에서 생성되는 신호값의 확률값을 그대로 사용하므로서 대상회로의 기능적인 활성도를 높였다. 그러므로 생성된

가중화 테스트벡터로서 대상회로의 고장 시뮬레이션 결과 신뢰성높은 로보스트의 테스트벡터의 수가 크게 증가하였으며 ISCAS85 벤치마크회로의 실험결과 전체적으로 평균 1.5~28배이상의 로보스트 고장 검출의 향상을 보였다. 이는 순수 LFSR로서 테스트를 발생시킨 경우와 볼때 절반이상의 ISCAS85 회로에서 약 50만개이상의 패턴을 인가해야지만 얻는 결과로 특히, 시스템 클럭 주기 결정에 영향을 주는 긴 경로들에 대해 보다 많은 수의 테스트벡터가 발생됨을 확인할 수 있었다. 또한, 주입력에 확률로서 할당된 논리값이 전파되면서 보다 많은 경로들을 활성화시킬 수 있도록 팬아웃지점에서의 균등한 신호전파 계산 알고리듬과 최장길이에 가까운 경로들을 로보스트하게 테스트할 수 있는 레벨차이에 따른 가중값 할당방식을 제안하였다. 또한 제안된 테스트방식은 IEEE 1149.1표준안에 적용될 수 있다.

앞으로의 연구과제로는 다중 가중값 집합(multiple weight set)의 생성과 이를 효과적으로 발생시킬 수 있는 레지스터 설계 방식에 관하여 연구하는 것이다.

참 고 문 헌

[1] Gordon L. smith, "Model for delay faults based upon path," *Proc. Int. Test* Conf., pp342-349, 1985.

[2] Keerthi Hragu, Michael L.Bushnell and Vishwani D.Agrawal, "An Efficient Path Delay Fault Coverage Estimator," *DAC* , pp516-521, 1994.

[3] I.Pomeranz, L.M.Reddy, "An efficient non-enumerative method to estimate path delay fault coverage," *ICCAD*, pp560-566 , 1992.

[4] J.A.Waicukauski and E.Lindbloom, "A method for generating weighted random test patterns," *IBM. J. Res. Dev.*, 33(2), pp149-161, March 1989.

[5] J.A.Waicukauski and E.Lindbloom, "Fault detection effectiveness of weighted random patterns," *Proc. Int. Test Conf.*, pp236-244 , 1988.

[6] P.H.Bardell, W.H.McAnney and J.Savir, *BUILT-IN TEST for VLSI : Pseudo-random Techniques.*, Wiley Interscience, 1987.

[7] F.Brglez, C.Gloster and G.Kedem, "Hardware-based weighted random pattern generation for boundary scan," *Proc. Int. Test Conf.* , pp264-273, 1989.

[8] F.Muradali, V.K.Agarwal and B.Nadean-Dosite, "A new procedure for weight random built-in self-test," *Proc. Int. Test Conf.*, pp660-669, 1990.

[9] Rohit Kapur, Srinvas Patil, Thomas J.Snethen and T.W.Williams, "Design of efficient weighted random pattern generation system," *Proc. Int. Test Conf.*, pp491-500, 1994.

[10] I.Pomeranz and S.M.Reddy, "3-weight pseudo random test generation based on a deterministic test set for combinational and sequential circuit," *IEEE Trans. Computer-Aided Design*, 12(7), pp1050-1058, July 1993.

[11] M.Bershteyn, "Calculation of multiple sets of weights for weighted random testing," *Proc. Int. Test Conf.*, pp1031-1040, 1993.

[12] Fardad Siavosh, "WTPGA : A novel weighted test-pattern generation approach for VLSI built-in self-test," *Proc. Int. Test Conf.* , pp256-262, 1988.

[13] Hans-Joachim Wunderlich, "Multiple distribution for biased random test patterns," *Proc. Int. Test Conf.*, pp236-245, 1988.

[14] F.Brglez, C.Gloster and G.Kedem, "Built-in self-test with weighted random pattern hardware," *ICCD.*, pp161-166, 1990.

[15] Clay S.Gloster and Franc Brglez, "Boundary scan with built-in self-test, " *IEEE Design and Test of computers*, Feb., 1989.

[16] F.Brglez and H.Fujiwara, Neutral netlist of ten combinational benchmark circuits and a target translator *in FORTRAN, in special session on ATPG and Fault simulation*, ISCAS, June 1985.

[17] Franz Fink, Karl Fuchs and Michael M.Schulz, "Robust and nonrobust path delay fault simulation by parallel processing of patterns," *IEEE Trans. Computers*, Vol 41, 12 pp1527-1536, Dec. 1992 .

[18] 허용민 외 , "조합 논리 회로의 경로 지연 고장 검출을 위한 효율적인 가중화 임의 패턴 생성 방법," 대한전자공학회 CAD 및 VLSI 설계 연구회 학술발표회 논문집, pp65-68, 1995

저 자 소 개

許 龍 珉(正會員) 第 31卷 A編 11號 參照
현재 한양대학교 전자공학과 대학원 박사과정

林 寅 七(正會員) 第 31卷 A編 11號 參照
현재 한양대학교 전자공학과 교수