

회로 분할에 의한 순차회로의 테스트생성

(Test Generation for Sequential Circuits Based on Circuit Partitioning)

崔 溟 鎔 *

(Ho-Yong Choi)

요 약

본 논문에서는 종래의 비명시적 적기계탐색법(IPMT법)의 적용한계를 개선하고자 회로 분할을 이용한 대규모 순차회로에 대한 테스트 생성법을 제안한다. 테스트 대상회로를 단일 출력을 가진 부분 회로로 분할하고, 부분 회로의 연결 정도를 나타내는 상태전이 코스트를 이용해 부분 스캔화 한다. 이 분할된 부분 스캔회로에 비명시적 적기계탐색법을 적용하여 테스트생성을 한다. IPMT법에서 테스트 생성이 불가능하였던 50개 이상의 플립플롭을 가진 ISCAS89 벤치마크 회로에 대한 실험 결과, 34%~73%의 스캔 플립플롭으로 거의 100%의 고장검출률로 고속으로 테스트생성이 가능하였다.

Abstract

In this paper, we propose a test generation method for large scale sequential circuits based on circuit partitioning to increase the size of circuits that the implicit product machine traversal (IPMT) method can handle. Our method partitions a circuit under test into subset circuits with only single output, and performs a partial scan design using the state transition cost that represents a degree of the connectivity of the subset circuit. The IPMT method is applied to the partitioned partial scan circuits in test generation. Experimental results for ISCAS89 benchmark circuits with more than 50 flip-flops show that our method has generated test patterns with almost 100% fault coverage at high speed by use of 34%~73% scanned flip-flops.

I. 서 론

최근의 반도체 제조 기술의 진보에 따라 논리회로의 집적도가 높아지고, 이에 따라 제조된 집적회로의 신뢰성 확보를 위한 고장검사(테스트)의 중요성이 더욱 높아지고 있다^[1].

논리회로의 고장검사는 회로의 외부입력에 입력계열을 인가하고, 이때 외부출력에 얻어지는 관측값과 회

로가 정상적으로 동작할 때에 출력되는 정상값과를 비교하여 검사한다. 이때 인가된 입력계열을 테스트계열(혹은 테스트)이라 하고 이 테스트계열을 생성하는 것을 테스트계열생성(test pattern generation: 이하 테스트생성이라 함)이라 한다.

조합논리 회로에 대한 테스트생성은 D 알고리즘^[2], PODEM^[3], FAN^[4], SOCRATES^[5] 등 여러 방법이 제안되어, 실용 회로 대부분의 고장에 대해 현실적인 시간으로 테스트 생성이 가능하다. 그러나, 순차논리 회로는 테스트생성에 방대한 상태공간을 탐색하는 것이 필요하고, 실용성의 문제에서 생기는 시간적 공간적 제약으로 탐색이 도중에 중단되는 고장이 자주 발생한다. 이 결과 지금까지 제안된 순차회로용

* 正會員, 忠北大學校 電氣電子工學部

(School of Electrical & Electronic Eng., Chungbuk National University)

接受日字:1997年7月19日, 수정완료일:1998年4月4日

의 테스트생성법은 소규모 혹은 중규모 정도의 회로에 대해서만 테스트생성이 가능하였다^[6-9]. 이 중 비명시적 적기계탐색(implicit product machine traversal: IPMT)법은 중규모의 회로를 대상으로 리던던트(redundant)가 아닌 고장에 대해서는 반드시 테스트가 생성된다는 의미에서 완전한 알고리즘으로 알려져 왔다^[8,9].

한편, 테스트 생성이 곤란하거나 불가능한 순차회로에 대해서는 스캔 설계(scan design)^[11] 등의 테스트 용이화 설계(design for testability) 방법을 이용할 수가 있다. 스캔 설계란 회로내의 기억소자(플립플롭) 모두를 연결해 하나의 쉬프트 레지스터(shift register)로 스캔패스화하여, 테스트 시에는 스캔패스를 통해 직접 플립플롭의 기억치를 쓰고 읽기 가능하도록 한 설계법이다. 이 설계법을 이용한 순차회로는 조합회로와 같이 취급되어 테스트생성을 용이하게 할 수 있다. 그러나 스캔 설계를 적용하면 회로 면적의 증대나 동작 속도의 저하를 가져오게 되어, 근래에는 최소한의 플립플롭만을 스캔 패스화 하여 회로의 성능 저하를 경감하는 부분스캔 설계법이 제안되어 있다^[10,11].

본 논문에서는 비명시적 적기계탐색법(IPMT법)에, 회로 분할을 이용하여 IPMT법의 적용한계를 개선하는 대규모 순차회로에 대한 테스트 생성법을 제안한다. 테스트 대상회로를 회로의 접속 그래프로 변환하여, 단일 출력을 가진 부분 회로로 분할한다. 분할된 회로에, 보다 소규모의 부분 회로로 분할되도록 부분 회로의 연결 정도를 나타내는 상태천이코스트를 이용하여 부분 스캔화 한다. 회로 분할을 이용한 부분스캔 설계를 하면 다음 상태 열거에 필요한 기억량이 현실적인 양으로 제한되어, 기존의 IPMT법을 이용하여 테스트생성이 불가능한 대규모 순차회로에 대해 효율적으로 테스트생성을 할 수 있다.

II에서는 비명시적 적기계탐색법에 관해 설명하고, III에서는 본 논문에서 제안하는 회로 분할과 부분스캔을 이용한 테스트생성에 관해 기술한다. IV에서는 ISCAS89 벤치마크 회로에 대한 실험결과를 보이고, 마지막 V에서는 결론과 금후의 과제에 대해 기술한다.

II. 비명시적 적기계탐색법

본 장에서는 본 논문에서 취급하는 회로 및 고장 모델을 기술하고 비명시적 적기계탐색법에 관해 설명

한다.

1. 회로 및 고장 모델

본 논문에서 사용하는 순차회로는 그림 1과 같이 조합회로부(CL)와 상태기억부(SR)로 구성되고, 외부 입력 n 개, 외부출력 m 개, FF l 개를 갖고 리세트 상태를 설정할 수 있는 동기식 순차회로이다.

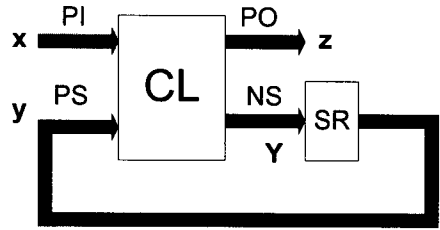


그림 1. 동기식 순차회로

Fig. 1. A synchronous sequential circuit.

외부입력 변수 벡터 x 와 현재상태 변수 벡터 y 를 각각 식(1)로, 리세트상태 r 를 식(2)로 표기한다.

$$x = (x_1, x_2, \dots, x_n), \quad y = (y_1, y_2, \dots, y_l) \quad (1)$$

$$r = (r_1, r_2, \dots, r_l) \quad (2)$$

각 외부출력 z_i 의 출력함수와 다음상태 Y_j 의 다음상태함수(혹은 상태함수)를 각각 식(3)과 식(4)로 표기하고, 이들도 마찬가지로 각각 식(5)로 벡터 표기한다.

$$z_i = f_i(x, y), \quad i = 1, 2, \dots, m \quad (3)$$

$$Y_j = g_j(x, y), \quad j = 1, 2, \dots, l \quad (4)$$

$$f = (f_1, f_2, \dots, f_m), \quad g = (g_1, g_2, \dots, g_l) \quad (5)$$

고장 모델은 게이트 수준(gate level)에서의 단일고착(stuck-at) 고장 모델을 사용한다. 고장회로에서의 변수 및 함수는 정상회로에서의 기호에 프라임(')을 붙여 y', f_i' 와 같이 표시한다. 외부출력의 故障差함수 D_F 와 다음상태의 故障差함수 D_G 를 각각 식(6)과 식(7)로 정의한다.

$$D_F(x, (y, y')) = \bigoplus_{i=1}^m [f_i(x, y) \oplus f_i'(x, y')] \quad (6)$$

$$D_G(x, (y, y')) = \bigoplus_{i=1}^l [g_i(x, y) \oplus g_i'(x, y')] \quad (7)$$

이것은 고장의 영향이 외부출력에 나타났을 때는 D_F 가, 다음상태 출력함수에 나타났을 때는 D_G 가 참으로 되는 함수이다.

積機械(product machine)는 그림2와 같이 정상회로(good machine)와 고장회로(faulty machine)가 같은 입력을 갖고, 각 상태의 積을 상태로 하고 고장차함수 D_F 를 출력으로 하는 기계이다. 이 적기계의 상태를 정상-고장상태쌍(이하 상태쌍)이라 부르고, $\theta=(y, y')$ 와 같이 표기한다. 입력 x 에 대하여 $D_F=1$ 을 만족하는 상태쌍 (y, y') 가 존재하면 이 상태쌍 (y, y') 를 불일치(inconsistent) 상태쌍이라 한다. 적회로에서 리세트상태 (r, r') 로부터 시작하여 시각 t 에 처음으로 도달하는 상태쌍집합을 θ_t 로 표시하고 이것을 시각 t 에 있어서 첫도달가능상태쌍 집합이라 한다.

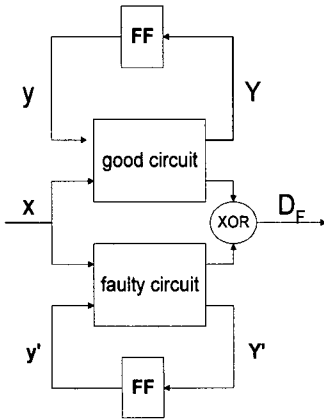


그림 2. 적기계
Fig. 2. A product machine.

2. 非明示的 積機械探索法

非明示的 積機械探索(implicit product machine traversal: IPMT)법^[8,9]은 積機械상에서 리세트상태로부터 도달가능상태쌍을 비명시적(implicit)으로 열거하여, 불일치상태쌍을 너비우선(breadth-first) 방식^[8] 혹은 너비우선/깊이우선(breadth/depth-first) 방식^[9]으로 탐색하는 테스트생성법이다. 이 방법은 적기계의 모든 상태쌍에 대해 탐색하기 때문에, 리던던트(redundant)가 아닌 고장에 대하여 반드시 테스트생성이 가능하다는 의미에서 완전한 알고리즘이다. 이 방법은 테스트생성 시간이 종래의 경로 활성화에 의한 테스트생성 알고리즘에 비해, 테스트생성이 쉬운 고장에 대해서는 효율이 약간 떨어지나, 테스트 곤란한 고장이나 리던던트 고장에 대해서는 고속으로 처리할 수 있어 유용하다.

IPMT에서 시각 t 에서의 탐색은 다음과 같이 수행한다.

- 1) 만약 θ_t 에 불일치상태쌍의 존재를 탐색한다. 존재하면 주어진 고장은 검출되고 탐색을 종료한다.
- 2) 존재하지 않으면 θ_t 로부터 θ_{t+1} 를 구한다.
- 3) $t=t+1$ 로 하고 다음 시각의 탐색을 한다.

IPMT법에서는 θ_t 와 같은 상태 집합을 특성함수(characteristic function)^[12]를 이용하여 비명시적으로 표현하고 상태 집합 연산을 논리함수로 처리한다. 또한 데이터 구조로 BDD(binary decision diagrams)^[13]를 이용하여 논리함수를 효율적으로 표시하고 연산을 고속으로 수행할 수 있다.

그러나 이 방법을 이용하여도 회로 규모가 어느 정도 크게 되면 특히 2)의 다음상태 열거처리(image computation: 상계산)에서 표현에 현실적인 허용량을 초과할 정도로 방대한 기억량이 필요하게 되고, 이 때문에 높은 검사율을 얻을 수가 없게 되는 문제가 있다.

III. 회로 분할에 의한 순차회로의 테스트 생성

IPMT법에 의한 대규모 순차 회로에 대한 테스트는 다음상태 열거처리(image computation: 상계산)에 방대한 기억량을 필요로 하여 테스트생성이 불가능하다. 따라서 상계산량을 줄이기 위해 본 논문에서는 회로 분할과 부분 스캔을 적용하여 테스트 생성을 한다. 먼저 회로를 단일 출력의 부분 회로로 분할하여 대상 회로의 규모를 줄인다. 그러나 이것만으로는 IPMT법을 그대로 적용하기 어려워 보다 소규모인 부분 회로로 분할하기 위해 부분 스캔법을 채택한다. 스캔된 플립플롭은 외부 입출력으로 간주되므로, 부분 회로의 규모가 더욱 제한된다. 이와같이 분할된 회로는 상계산량이 줄어들어 IPMT법을 적용할 수 있어 효율 좋은 테스트 생성이 가능하다. 본 장에서는 먼저 회로 분할 방법에 관해 기술하고, 후에 이를 이용한 테스트 생성에 관해 기술한다.

1. 회로 분할

회로 분할을 설명하기 위해 용어를 다음과 같이 정의한다.

회로 연결그래프

외부입력(PI), 외부출력(PO), 플립플롭(FF)으로 구

성된 회로 C 의 회로구조상의 접속관계를 유향그래프 (directed graph) $G_c = (V_c, E_c)$ 로 정의한다. $V_c = V_c^{pi} + V_c^{po} + V_c^{ff}$ 는 노드(node)의 집합, E_c 는 유향 에지 (directed edge)의 집합이다. 회로 상의 외부입력, 외부출력, 플립플롭은 각각 G_c 상의 외부입력 노드 $v \in V_c^{pi}$, 외부출력 노드 $v \in V_c^{po}$, 플립플롭 노드 $v \in V_c^{ff}$ 와 일대일로 대응된다. 에지 $e = (u, v) \in E_c$ 는 C 상에서 노드 u 에 대응하는 소자로부터 조합논리 부를 1회 통해 노드 v 에 대응하는 소자로의 경로가 존재할 때 이 경로에 대응된다.

그림 3의 예를 보자. 주어진 테스트회로가 주어지면 회로접속 그래프 G_c 는 그림 (a)와 같이 외부입력은 노드 ▲로, 외부출력은 노드 △로, 플립플롭은 ●로 표현되고 이들 사이에 있는 조합회로부분은 입력에서 출력으로 향하는 단지 유향에지로 표현된다.

단일 출력 부분그래프

회로의 연결그래프 G_c 에서 외부출력 p 에 대한 단일 출력 부분그래프(subset graph with single output) G_p 는, 그래프 G_c 로부터 p 에 도달되지 않는 노드를 모두 제거하여 얻어지는 그래프이다.

그림 3의 예에서 외부출력 p 에 대한 단일 출력 부분그래프 G_p 는 그림 3(a)의 회로 연결그래프 G_c 에서 p 에 대해 p 에 도달되지 않는 노드를 모두 제거하여 얻어진 그래프로 그림 3(b)와 같이 된다.

상태전이 코스트

외부출력 p 에 대한 단일 출력 부분 그래프 G_p 로부터 노드 p 를 제거한 그래프를 G_{p-} 라 하자. 이때 G_p 의 상태전이 코스트(state transition cost)는 G_{p-} 의 連結 성분 중에 포함되는 플립플롭 노드의 최대수로 정의한다.

그림 3의 예에서 p 를 제거한 그래프 G_{p-} 는 그림 3(c)로 되고 상태전이 코스트는 3이 된다.

이상의 정의를 이용하여 회로의 분할은 다음과 같이 정의된다. 회로 C 의 연결그래프 G_c 의 분할은

$$\Pi(G_c) = (G_{p1}, G_{p2}, \dots, G_{pm})$$

로 된다. 이때 회로는 각각 G_{pi} 에 대응하는 단일출력 부분 회로로 분할된 회로이다.

테스트생성 시에는 분할된 각 부분 회로에 대하여 IPMT법을 적용한다. 다음상태 열거는 각각의 연결 성분에 포함되는 플립플롭의 노드에 대응한 부분 상태에 상태를 서로 변수 의존이 없는 부분 상태로 분해하여, 부분 상태마다 다음상태를 열거하고, 얻은 상태 집합들의 곱으로서 본래의 상태 집합을 얻는다. 즉, 회로의 플립플롭에 도달 가능한 부분을 추출하여, 그 중에 連結인 부분에 포함되는 플립플롭에 대응한 부분 상태에, 상태를 직적 분해(product decomposition)하고, 각 부분 상태마다 다음상태를 열거(상계산)하는 것이다.

그러나 상기의 회로분할만으로는 대규모 회로에 대해서는 아직도 방대한 기억량이 필요하게 되어 IPMT법을 직접 적용하기에는 불가능하다. 따라서 IPMT법을 적용할 수 있도록 보다 회로를 소규모로 할 필요가 있고 이를 위해 다음 절의 부분 스캔법을 이용하도록 한다.

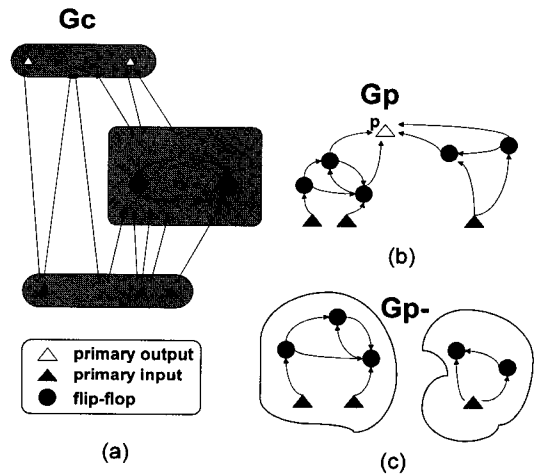


그림 3. (a) 회로 연결그래프 (b) 단일 출력 부분 그래프 G_p (c) G_{p-}

Fig. 3. (a) A circuit connected graph (b) A subset graph with single output G_p (c) G_{p-}

2. 부분 스캔을 이용한 회로 분할

III.1의 단일 출력 부분회로로의 회로 분할만으로는 테스트생성 시 소요 기억량의 삭감이 불충분하므로 회로를 복수의 보다 소규모인 부분 회로로 분할하기 위해 부분 스캔법을 채택한다. 스캔된 플립플롭은 외부 입출력으로 간주되므로, 부분 회로의 규모를 더욱 더 제한할 수 있고, 따라서 상계산을 용이하게 할 수

있어 IPMT법을 적용하여 테스트 생성을 할 수 있다.

회로규모의 척도로서 상태전이 코스트(state transition cost)를 이용한다. 또한 회로의 순차적 심도(sequential depth: 연결그래프 상에서의 외부출력으로부터 외부입력까지의 거리)는 테스트 길이에 영향을 미치고, 심도가 작은 쪽이 테스트 길이가 짧게 될 수 있으므로, 스캔 플립플롭의 선택에는 외부출력으로부터 입력 측으로 너비우선 탐색(breadth-first traversal)을 이용한다.

이 스캔 플립플롭 선정을 위한 알고리즘은 그림 4와 같다. 입력은 회로 C 와 상태전이 코스트의 상한치 K 이다.

Step 1. 회로의 접속 그래프 G_c 를 추출한다.

Step 2. 각 외부출력에 대하여

Step 2-1. G_p- 를 구한다.

Step 2-2. G_p- 의 각 연결 성분에 대하여 제약을 만족하도록 FF의 스캔을 행한다(상태전이코스트 K).

Step 2-2-1. 큐(queue)를 공집합으로 하여 초기화한다. 모든 노드를 미방문으로 하고 $k=0$ 으로 한다.

Step 2-2-2. p 에 인접한 노드를 모두 이미 방문한 것으로 하고 큐에 삽입한다.

Step 2-2-3. $k > K$ 면 큐 내의 모든 FF의 노드를 스캔한다. 스캔에 의해 FF 노드는 입력 에지를 이어 받는 외부출력 노드점과 출력 에지를 이어 받는 외부입력 노드점으로 분리한다.

Step 2-2-4. 큐가 공집합이 되면 종료한다. 그렇지 않으면 큐로부터 노드 v 를 하나씩 꺼낸다.

Step 2-2-5. v 가 FF이면 $k=k+1$.

Step 2-2-6. v 로의 출력 에지를 갖는 미방문의 노드를 모두 큐에 넣는다. Step 2-2-3으로

Step 3. 과잉 스캔한 FF(상태전이코스트 K 보다 작은 경우)이 있으면 원상 복귀한다.

그림 4. 스캔 플립플롭 선정 알고리즘 sel_psff

Fig. 4. An algorithm sel_psff for selecting scan FFs.

먼저 주어진 회로에 대해 접속 그래프를 추출하고, 각 외부출력에 대해 회로를 분할하여 G_p- 를 얻는다. Step 2에서는 각 G_p- 에 대해서 상태전이 코스트의 상한치가 K 를 넘지 않도록 스캔 플립플롭을 너

비우선 탐색하여 선택한다. Step 3에서는 FF이 2개 이상의 분할회로에 속해 한 분할회로에서는 정상 FF으로 되고 다른 분할회로에서는 스캔 FF으로 되었을 경우 이 FF을 일단 과잉 FF으로 판정하고, 다시 스캔 FF여부를 판단하여 가능한 한 정상 FF으로 되돌려 스캔 FF 수를 줄인다. 줄일 때는 Step 2에서 스캔된 것과 역순으로, 스캔되어 외부 입력력 노드로 된 것을 FF 노드로 되돌리고 상태전이 코스트가 K 이하로 계속되는 가를 조사한다. K 보다 크게 되면 이 노드를 다시 스캔노드로 고친다.

이 방법을 이용하여 부분 스캔설계를 하면 다음상태 열거에 필요한 기억량이 현실적인 양으로 제한되어 회로 분할과 상태의 직적분해(product decomposition)를 채용한 IPMT법에 의해 효율적으로 테스트생성을 할 수 있다.

3. 회로 분할에 의한 순차회로의 테스트생성

III.1과 2의 단일 출력 부분회로의 회로 분할과 부분 스캔설계를 이용한 회로 분할로 회로의 규모를 더욱 제한하여 상계산을 용이하게 함으로써, IPMT법에 의해 테스트생성을 효율적으로 할 수 있다. 본 절에서는 이들 회로 분할에 의한 테스트생성에 관해 기술한다.

그림 5는 회로 분할에 의한 테스트생성 알고리즘을 나타낸다. 테스트 대상 회로에 대해 III.2의 부분 스캔 설계 알고리즘 sel_psff 를 이용하여 주어진 상태전이 코스트 K 에 대해 스캔할 플립플롭을 선정한다. 선정된 스캔 플립플롭은 외부 입력력으로 간주하여 테스트 회로를 부분 스캔된 회로로 변형하고, III.1에서 설명한 단일 출력 부분회로로 회로 분할을 한다. 이때의 단일 출력 부분회로는 상태전이 코스트가 K 이하로 되어 다음상태 열거에 필요한 기억량이 현실적인 양으로 제한되어 IPMT법^[9]을 적용할 수 있는 규모로 제한된다. 다음에 각 분할된 단일출력 부분회로의 고장 각각에 IPMT법을 적용하여 테스트생성을 한다. 이때 다음상태 열거는 분할한 부분회로를 고려하여 상태를 상호 변수의존이 없는 부분상태로 분해하고, 부분상태마다 다음상태를 열거하여 얻은 부분상태의 집합의 곱으로 본래의 상태집합을 얻는다. 즉, 부분회로 중에 連結인 부분에 포함되는 플립플롭에 대응한 부분 상태에, 상태를 직적 분해(product decomposition)하고, 각 부분 상태마다 다음상태를 열거한다.

- Step 1. Select partial scan FFs in a circuit under test according to *sel_poff* algorithm with state transition cost K.
- Step 2. Get a partial scanned circuit by replacing the outputs of scan FFs with primary inputs (PIs) and the inputs of the scan FFs with primary outputs (POs) in the given circuit.
- Step 3. Partition the scanned circuit into subcircuits with a single output.
- Step 4. Apply IPMT algorithm to each fault in partitioned subcircuits.

그림 5. 회로분할에 의한 테스트생성 알고리즘
Fig. 5. Test generation algorithm based on circuit partitioning.

IV. 실험 결과

III에서 기술한 회로분할에 의한 테스트생성 알고리즘을 C언어로 실현하고, SPARCstation2상에서 ISCAS89 벤치마크회로^[14] 중 플립플롭 50개 이상을 가진 대규모 회로(중전의 IPMT법에 의해서는 테스트생성이 불가능했던 회로)에 대해 실험하였다. 각 회로에 대해 회로 분할을 하고, 규모가 작은 부분회로부터 차례로 테스트 대상으로 하였다. 테스트생성에서의 논리함수의 처리는 문헌 [13]의 SBDD 연산 프로그램을 이용하였고 실행시 SBDD의 최대노드수는 2^{20} ($\approx 10^6$)으로 하였다.

실험에 이용한 ISCAS89 벤치마크 회로의 사양을 표 1에 나타냈다. 표 내의 #PI, #PO, #FF, #gate, #fault는 각각, 회로의 외부입력수, 외부출력수, 플립플롭수, 게이트수, 고장수를 나타낸다. 고장수는 고착고장 모델을 이용한 대표 고장이다. 모든 플립플롭의 상태가 0인 상태를 리셋 상태로 가정하였고 고장시물레이션은 사용하지 않았다. 테스트생성 결과는 표 2와 같다. 표의 각 항은 #node가 연결그래프의 노드수, #edge가 에지수, K가 천이 계산 코스트의 상한, #FF이 원래 회로의 플립플롭의 수, 이 스캔된 플립플롭의 수, 팔호안은 스캔율($\#FF/\#scan \times 100\%$), t. e.는 고장검출률, 시간은 테스트생성 시간(단위는 초)이다. 고장검출률은

$$\text{고장검출률} = \frac{\text{고장수} - \text{중단고장수}}{\text{고장수}} \times 100(\%)$$

이다(고장검출률에는 조합적 리던던트(combinationally redundant)고장과 순차적 리던던트(sequentially redundant) 고장이 모두 포함되어 있음).

5~20의 K 값으로, s38417 회로를 제외하고 스캔율 34%~73%의 부분 스캔회로에서 100%의 고장검출률을 얻었다. 또한 테스트생성이 실용적인 시간 내에 가능하였다. s9234 및 s13207에서는 K가 표의 값보다 크면 테스트 생성시간이 갑자기 더 길어진다. 이것은 부분회로 내에 카운터 회로 같은 것이 남아 매우 천이가 깊은 부분이 있기 때문으로 보인다. 본 부분스캔법의 실용화를 위해서는 이와 같은 부분이 남지 않도록 제약을 추가할 필요가 있다고 생각된다. 이 제약은 연결그래프 내의 사이클 길이를 제한하는 것에 의해 실현 가능하리라 본다. K의 값은 실험 대상 회로의 크기와 중간 실험 결과를 고려하여 정한 것으로, 일반적으로 FF의 개수가 많은 회로일수록 상계산량이 많아져 K의 값은 적게되는 경향이 있다.

메모리 사용량은 테스트생성 시 개별 고장에 대해 SBDD의 최대노드수²⁰으로 제한되며, s38417회로를 제외한 모든 회로에 대해 이 범위 내에 있어 테스트생성이 가능하였다. 그러나, s38417 회로의 일부 고장에 대해서는 노드수가 2^{20} 을 초과하여 100% 고장검출률을 얻을 수 없었다. 즉 이는 고장회로의 논리함수를 기억량의 제약으로부터 얻을 수 없었거나, 다음상태 열거(상계산)의 계산량이 과대하여 나타났기 때문이다. 이것을 피하기 위해서는 BDD의 변수 순서를 보다 기억 효율이 좋은 방법을 고안할 필요가 있다.

본 실험 대상인 ISCAS의 대규모 회로들은 기존의 IPMT법^[8,9,11]에서는 테스트생성이 불가능하였던 회로들이었다. 따라서 회로분할과 부분스캔에 의해 대규모 회로에 대해서도 IPMT법을 적용하여 효율적으로 테스트생성이 가능하다고 할 수 있다. 한편 다른 부분스캔법과의 비교는 타 부분스캔법이 스캔의 효율성만을 위하였고, 또한 테스트생성 알고리즘도 달라 비교하기 어려우나, 참고로 참고문헌 [10]에 기술된 s5357, s9234에 대한 결과와 비교하면 [10]의 스캔율은 각각 18%, 23%에 99.7%, 99.0%의 고장검출률을 얻는데 반해, 본 논문에서는 34%, 45%의 스캔율에 100%의 고장검출률을 얻어, 본 논문의 스캔율이 높고 고장검출률은 비슷하나 테스트생성 시간은 짧다고 할 수 있다.

표 1. ISCAS89 벤치마크회로

Table 1. ISCAS89 benchmark circuits.

name	#PI	#PO	#FF	#gate	#fault
s1423	17	5	74	753	1515
s5378	35	49	179	3042	4603
s9234	19	22	228	5866	6927
s13207	31	121	669	8722	9815
s35932	35	320	1728	18148	39094
s38417	28	106	1636	23949	31180
s38584	12	278	1452	20955	36303

표 2. 부분스캔회로의 테스트생성 결과

Table 2. Test generation results for partial scan circuits

name	#node	#edge	#FF	K	#scan	t. e.	time
s1423	96	4470	74	10	54(73%)	100	213
s5378	263	4623	179	10	95(53%)	100	55
				15	87(49%)	100	138
				20	61(34%)	100	2203
s9234	269	6520	228	10	102(45%)	100	316
				15	105(46%)	100	335
s13207	821	9442	669	5	338(51%)	100	816
s35932	2083	15190	1728	10	899(52%)	100	1356
s38417	1770	68702	1636	5	1037(63%)	87	29492
s38584	1742	40888	1452	5	956(66%)	100	2990

V. 결 론

본 논문에서는 외부출력을 중심으로 하고, 또한 부분스캔법을 이용한 회로분할을 기술하고, 분할된 회로에 IPMT법을 적용한 테스트생성법을 제안하였다. 또한 ISCAS89벤치마크 회로에 대한 실험을 통해 이의 테스트생성 결과를 고찰하였다. IPMT법에 의해 원래의 회로 자체로는 거의 테스트생성을 할 수 없었던 50개 이상의 플립플롭을 가진 ISCAS89 벤치마크 회로에 대한 실험 결과, 34%로부터 73%의 플립플롭 스캔에 의해 실용적인 시간 내에 거의 테스트생성을 할 수가 있었다. 본 방법의 이점은 테스트생성에 있어서 한번에 취급할 회로의 규모를 제한하여, 테스트생성의 효율을 높일 수 있다는 점이다. 또한, 회로가 분할되기 때문에 테스트 생성에 병렬처리가 쉽게 도입할 수도 있다.

금후의 과제로는, 보다 더 상세하게 회로 규모를 평가하여 보다 적은 플립플롭의 스캔률로 테스트할 수 있는 테스트생성법과 보다 고속 처리를 위해 병렬처리에 의한 테스트생성법 등이 고려된다.

참 고 문 헌

- [1] M. Abramovichi, M.A. Breuer, and A.D. Friedman, Digital Systems Testing and Testable Design, Computer Science Press, New York, 1990.
- [2] J. P. Roth, "Diagnosis of automata failures: A calculus and a method," IBM J. of Research and Development, vol. 10, pp. 278-291, July 1966.
- [3] P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits," IEEE Trans. Computer, vol. C-20, no. 3, pp. 215-222, Mar. 1981.
- [4] H. Hujiwara and T. Shimono, "On the acceleration of test generation algorithm," IEEE Trans. Computer, vol. C-32, no. 12, pp. 1137- 1144, Dec. 1983.
- [5] M. H. Schultz, E. Trischler, and T.M. Sarfert, "SOCRATES: A highly efficient automatic test pattern generation system," IEEE Trans. on Computer-Aided Design, vol. CAD-7, no. 1, pp. 126-137, Jan. 1988.
- [6] W. T. Cheng and T.J. Chakraborty, "Gentest: An automatic test generation system for sequential circuits," IEEE Computer, vol. 22, no. 4, pp.43-49, Apr. 1989.
- [7] A. Ghosh, S. Devadas, and A.R. Newton, "Test generation and verification for highly sequential circuits," IEEE Trans. on Computer-Aided Design, vol. CAD-10, no. 5, pp. 652-667, May 1991.
- [8] H. Cho, G.D. Hachtel, and F. Somenzi, "Fast sequential ATPG based on implicit state enumeration," in Proc. Int. Test Conf., pp. 67-74, Oct. 1991.
- [9] 崔漢鎔, 小原, 石浦, 白川, 本原, "論理關數處理に基ついた順序回路のテスト生成法", 電子情報通信學會論文誌, vol. J76-A, no. 6, pp. 835-843, 1993年 6月 (in Japanese).
- [10] K.T. Cheng and V.D. Agrawal, "A partial scan design for sequential circuits with feedback," IEEE Trans. Computer, vol. C-39, no. 4, pp. 544-558, Apr. 1990.

- [11] 최 호용, “논리합수처리에 의한 부분스캔순차회로의 테스트생성,” 정보처리학회 논문지 제3권 제3호, 한국정보처리학회, pp. 572 - 580, 1996년 5월
- [12] H. J. Touati, H. Savoj, B. Lin, R. K. Brayton, and A. Sangiovanni-Vincentelli, “Implicit state enumeration of finite state machines using BDD’s,” in Proc. IEEE Int. Conf. Computer-Aided Design, pp. 130-133, Nov. 1990.
- [13] S. Minato, N. Ishiura, and S. Yajima, “Shared binary decision diagram with attributed edges for efficient Boolean function manipulation,” in Proc. 27th Design Automation Conf., pp. 52-57, June 1990.
- [14] F. Brglez, D. Bryan, and K. Kozminski, “Combinational profiles of sequential benchmark circuits,” in Proc. Int. Symp. on Circuits and Systems, pp. 1929-1934, June 1989.

 저 자 소 개



崔 溟 鎔(正會員)

1957년 2월 1일생. 1980년 서울대학교 전자공학과 졸업(공학사). 1982년 한국과학기술원 전기 및 전자공학과 졸업(공학석사). 1994년 오호사카대학 대학원 전자공학과 졸업(공학박사). 1982년 ~ 1985년 삼성반도체(주) 연구소 연구원, 선임연구원. 1985년 ~ 1986년 부경대학교 전자공학과 전임강사, 조교수, 부교수. 1996년 ~ 현재 충북대학교 전기전자공학부 부교수. 주관심분야는 집적회로 테스트 및 설계