論文2002-39SD-9-4

다중 전송선에 영향을 받는 Crosstalk 잡음을 위한 테스트 생성

(Test Generation for Multiple Line Affecting Crosstalk Effect)

李 永 均 *, 梁 善 雄 *, 金 文 俊 *, 張 勳 **

(Young-Gyun Lee, Sun-Woong Yang, Moon-Joon Kim, and Hoon Chang)

요 약

VLSI 영역에서 전송선에 발생하는 상호교차 커패시턴스(cross-coupling capacitance)가 중요한 이슈가 됨에 따라 이로 인한 고장을 검출하는 몇 가지 ATPG 알고리즘이 제안되었다. 대부분 단일한 능동선로만을 대상으로 연구가 진행되었으며, 테스트 생성 효율에 비해 많은 시간비용을 감수해야 하는 결과를 내 놓을 수 밖에 없었다. 이에 대한 대안으로 본 논문에서는 다중 선로를 대상으로 하는 잡음 모델에 관해 연구하였다. 본 논문은 다수의 전송선에 영향을 받는 crosstalk 모델을 제시하고 이 모델에 따라 crosstalk 잡음 고장 검출을 목적으로 하는 ATPG 알고리즘을 제안한다. 이 논문에서는 crosstalk에 의한 잡음 고장을 정적 해저드로 모델링하고 있으며 비용함수는 적용하지 않는다. 또한, 다중 전송선에 의한 crosstalk 잡음을 발생시키는 논리값 조건을 설정하고 각 게이트에 따라 이 조건을 만족하는 진리표를 만들 것이다. 그 후 PODEM에 기반한 ATPG 알고리즘을 구현한 후 그 결과를 보인다.

Abstract

As cross-coupling capacitance generated in transmission line has been an important issue in VLSI world, a couple of ATPG algorithms has been proposed. However they were studied only for a simple single-line effect problem, so it cost so much time for an unsatisfying test generation efficiency. In this paper, we studied a noise model for multiple affected lines and generated test patterns in a short time. This paper proposes a crosstalk model affected by multiple tranmission lines and implemented an ATPG algorithm for detection of crosstalk noise faults. We modeled the crosstolk noise fault as a static hazard and not applying a cost function. We set a logic condition generating a crosstalk noise by multiple transmission line and made a truth table for this. We implemented an ATPG algorithm based on PODEM and revealed the results.

Ⅰ. 서 론

GHz 범위의 클럭속도와 딥 서브마이크론(deep submicron) 공정을 이용하는 VLSI 시스템에서 시스템 성능에 비례해 증가하는 상호교차 커패시턴스(cross-coupling capacitance)는 정상적인 회로기능과 시스템 성능에 반하는 중요한 문제점이다. 설계검증 과정에서 이에 관한 충분한 고려가 없다면 crosstalk는 신호지연 또는 논리 해저드와 같은 예기치 못한 고장의 원인이 될 수 있다.[1]

칩 면적과 성능 제한이 심하지 않은 경우라면 설계검증 과정에서 발견된 오류는 설계를 수정하거나 신호배선을 변경함으로써 제거 가능하나,[2] 고성능을 목적

* 正會員, 崇實大學校 컴퓨터學科
(Department of Computing, Graduate School, Soongsil University)
** 正會員, 崇實大學校 컴퓨터學部
(School of Computing, Soongsil University)
接受日字:2001年12月3日, 수정완료일:2002年8月19日

으로 하는 설계에서 회로 내에 존재하는 모든 잡음을 제거하는 것은 불가능한 일이다. 따라서, 완성된 칩의 정확한 동작을 위해서는 잡음현상에 대한 엄격한 테스트가 필요하다.

RC 선로 상에서 crosstalk로 인한 잡음을 검출 및 해석하는 방법이 [3][4]에 기술되어 있다. 또한 crosstalk를 위한 테스트 패턴 생성에 관한 연구도 최근에 시작되었다. [5][6]은 논리 수준의 crosstalk 고장 모델에 대하여 연구하였으며 각각 PODEM과 FAN에 기반한 ATPG(Automatic Test Pattern Generation) 알고리즘을 제안하였다. 이들은 crosstalk에 의한 잡음 고장을 최고전압 스윙(full voltage swing)을 갖는 정적 해저드로 모델링하고 모델링된 crosstalk 잡음을 발생시키기 위한 주입력단의 입력벡터를 생성하는 알고리즘과 검출된 crosstalk 잡음을 주출력단으로 전파시키는 방법론에 중점을 두고 있다. [7]은 [5][6]의 연구에 비용함수(cost function)를 도입하여 잡음의 세기(strength)에 따라 불필요한 테스트 벡터는 생성하지 않도록 하였다. 그러나 [7]은 적정 크기의 잡음을 충족하는 경로를 찾기 위해 반복적으로 여러 개의 테스트를 생성하고, 이 과정에서 각 게이트마다 비용함수를 적용하여 수학적 연산을 수행하기 때문에 [5]에 비해 테스트 생성 시간이 상대적으로 많게 된다.

지금까지의 crosstalk 잡음을 위한 ATPG 연구에서는 2개의 전송선로 간에 발생하는 상호교차커패시턴스에 대해서만 모델링이 이루어져 왔다. 즉, 이웃한 하나의 전송선로에 영향을 받는 단일 피해선로에 대한 모델링이 그것이다. 그러나, crosstalk 잡음은 피해선로에 영향을 주는 전송선이 많아질수록 잡음의 크기도 커지는 특성을 가지기 때문에,[8] 단일선로에 비해 현실적인 모델링 방법이 될 수 있다.

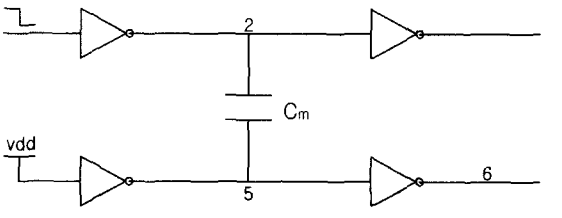
본 논문에서는 고집적, 고속 전송선에서 발생하는 crosstalk 잡음고장에 대한 빠르고 효율적인 테스트 환경 구축을 위해 다수의 전송선에 영향을 받는 crosstalk 모델을 제시하고 이 모델에 따라 crosstalk 검출을 목적으로 하는 ATPG 알고리즘을 제안한다. 이 논문에서는 crosstalk에 의한 잡음 고장을 최고전압 스윙을 갖는 정적 해저드로 모델링하고 있으며 비용함수는 적용하지 않는다. 또한, 다중 전송선에 의한 crosstalk 잡음을 발생시키는 논리값 조건을 설정하고 각 게이트에 따라 이 조건을 만족하는 진리표를 작성할 것이다. 그 후 PODEM[9]에 기반한 ATPG 알고리즘을 구현한 후 그 결과를 보인다.

본 논문의 구성은 다음과 같다. 2장은 다중 전송선에 영향을 받는 crosstalk 잡음 모델에 대해 논하고 있으며, 이 모델을 근간으로 3장에서는 crosstalk 잡음 고장을 검출하기 위한 ATPG 알고리즘을 제안하고 구현한다. 그리고 그 결과를 ISCAS 회로에 적용하여 성능 평가한 내용을 4장에 기술하였다. 마지막으로 5장에서 본 논문의 결론을 맺는다.

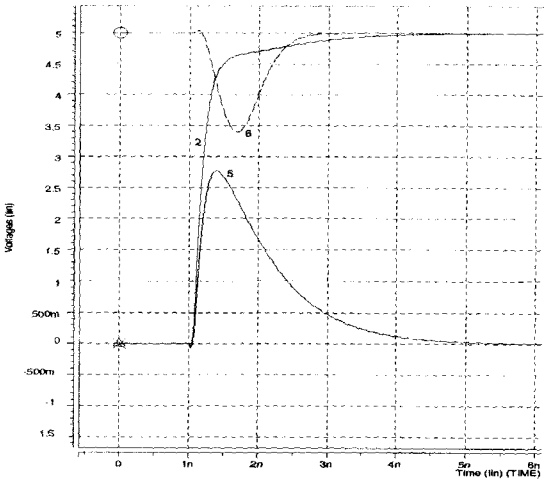
Ⅱ. 다중 선로에 영향을 받는 crosstalk 잡음 모델

1. 단일 능동선로에 의한 crosstalk 잡음

그림 1에 단일한 능동선로로 인해 일어날 수 있는 피해선로의 crosstalk 잡음 효과를 보였다. 그림 1 (b)는 모델회로(a)를 SPICE 시뮬레이션한 과도해석의 결과이다. 능동선로와 피해선로의 드라이버 크기는 대략 2:1의 비율을 가지며, 상호교차 커패시턴스 C_m은 0.3pF로 가정하였다. 노드 2에서 신호가 상향 또는 하향 트랜지션 할 때 노드 5에는 각각 상향잡음신호, 하향잡음신호가 발생된다. 그림 1 (a)의 경우 상향잡음신호는 직후 인버터의 논리값에 영향을 줄 수 있을 것이라 예상되는 문턱전압에 도달해 있다. 이 잡음신호는 그림 1 (b)의 시뮬레이션 결과에서와 같이 시험회로의 노드 6에서 논리 출력값 1에 잡음신호를 발생한다. 그러나 노드 6에서 발생한 잡음신호는 게이트를 구동하기엔 다소 미약한 신호이므로 노드 6 이후의 게이트에는 큰 영향을 미치지 않는다.[10] 다중선로를 사용하는 버스라인에서와 같이 피해선로 하나에 대해 그림 1에서 가정한 능동선로 여러개가 영향을 주는 경우를 다음 절에서 다룬다.



(a) 시험 회로
(a) Test circuit



(b) 시뮬레이션 결과
(b) Simulation result

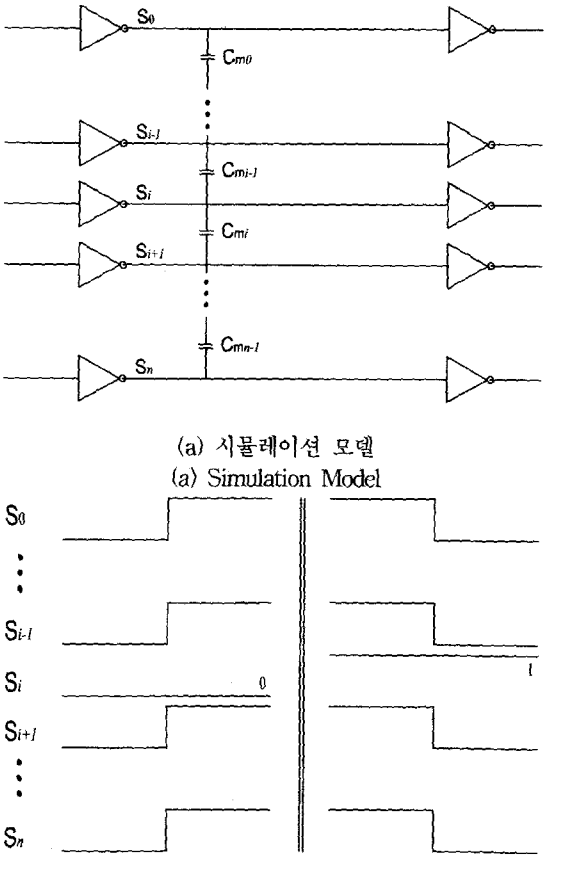
그림 1. 단일 능동선로에 의한 Crosstalk 잡음 효과
Fig. 1. Single line affecting crosstalk noise model.

2, 다중 능동선로의 영향

다중 능동선로에 의한 모델링은 피해선로에 단일선로의 경우보다 큰 상호교차 커패시턴스를 만든다. 즉, 고장영역에 최대 crosstalk 잡음을 발생하게 함으로써 좀 더 효율적인 테스트 벡터가 생성되도록 하였다. 다중 선로의 영향에 대한 모델링은 단일 선로로 모델링한 것에 비해 다음과 같은 두가지 이점이 있다. 첫 번째, 이 방법은 최대 잡음을 만들기 위해 회로 공간을 탐색하는 시간적 비용이 [7]의 경우보다 상대적으로 적다. 두 번째, 단일선로에 의한 경우를 포함하여 필요에 따라 많은 수로 이루어진 능동선로의 영향을 관찰 할 수 있는 유연성이다. 이 경우의 일례로, 다수의 전송선으로 이루어진 버스라인을 생각해 볼 수 있다. 일반적으로 단일 버스라인 상에서 발생하는 crosstalk 효과는 인접한 몇 개의 능동선로에 직접적으로 영향을 받는다. 이런 경우에 하나 또는 그 이상의 능동선로가 주는 영향을 비교, 관찰 할 수 있다.

다중 능동선로에 영향을 받는 crosstalk 모델을 그림 2에 나타내었다. 선로 S0에서 Sn이 있을 때, Si를 피해선로로 가정하고 나머지는 능동선로로 한다. 상호교차 커패시턴스는 C_m으로 표시하였다. 그림 2 (a)는 피해선로 Si와 n-1개의 능동선로 간에 영향을 주고받는 모두 n-1개의 상호교차 커패시턴스가 존재하고 있음을 나타내고 있다. 이 상황에서 Si에 정적인 논리값 0이 흐르고 있을 때 능동선로에 상향 트랜지션 신호를, 반대로 Si에 정적인 논리값 1이 흐르고 있을 때 능동선로에 하향 트랜지션 신호를 인가하면 피해선로 Si에 각각 상향잡음신호, 하향잡음신호가 생성된다. 다중 능동선로에 필요한 테스트패턴은 그림 2 (b)에 나타나 있으며, 왼쪽과 오른쪽에 피해선로의 논리값이 각각 0과 1일 경우를 구분하였다.

예를 들어, 두 개의 능동선로에 영향을 받는 피해선로가 있을 때, 능동선로와 피해선로 모두에 정적인 논리값 0이 흐르고 있다고 가정하자. 그 후, 능동선로의 논리값이 1로 바뀌면 피해선로에 잡음이 발생한다. 피해선로에 영향을 주는 커패시턴스의 크기는 각각의 능동선로가 피해선로와 관계하는 커패시턴스의 합과 같으며, 그림 3에 이러한 상황을 나타내었다. 능동선로와 피해선로의 드라이버 크기는 대략 2:1의 비율을 가지며, C_{m1}과 C_{m2}는 각각 0.3pF로 가정하였다. 따라서, 피해선로인 노드 5에 영향을 주는 상호교차 커패시턴스의

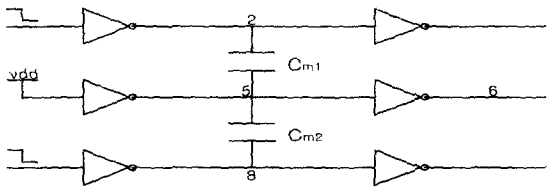


(a) 시뮬레이션 모델
(a) Simulation Model

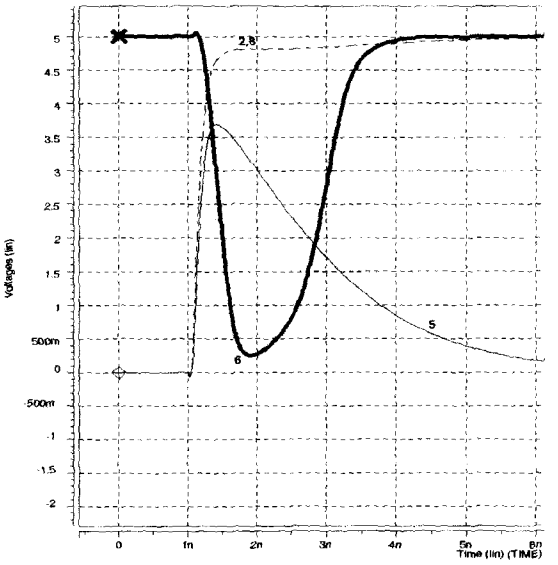
(b) 테스트 패턴
(b) Test Pattern

그림 2. 다중 능동선로에 영향을 받는 crosstalk 모델
Fig. 2. Multiple line affecting crosstalk noise model

크기는 0.6pF으로써 직후 인버터의 문턱전압을 초과하여 노드 6에도 잡음이 전달되는 것을 볼 수 있다. 이 잡음신호는 이후 연속된 게이트를 구동하여 지난 절에서 기술한 단일한 능동선로에서의 예와 달리 주출력단까지 전파된다.



(a) 시험 회로
(a) Test circuit



(b) SPICE 시뮬레이션 결과
(b) SPICE Simulation result

그림 3. 다중 능동선로에 영향을 받는 crosstalk의 예
Fig. 3. Example of Multiple line affecting crosstalk.

Ⅲ. 테스트 생성기

이번 장에서는 crosstalk 잡음을 위한 테스트 패턴 생성 알고리즘에 대하여 다룬다. 이 알고리즘은 PODEM 알고리즘에 기초를 두면서 다중 선로를 위한 테스트 생성이 가능토록 한 개량형 PODEM 알고리즘이다. 이 테스트 생성기는 지정된 crosstalk 잡음선로에 잡음을 발생시킬 수 있는, 즉 직후 게이트에서 논리 에러를 유발할 수 있는 벡터집합을 생성하고, 유발된 잡음을 주출력단에서 감지할 수 있도록 전파시키는 것을 목적으로 한다.

1. 사용된 논리값과 논리값에 따른 회로동작

표 1에 본 논문에서 사용하고 있는 논리값에 대해 간략히 설명하였으며, 그림 4에는 이 논리값을 사용한 기본 게이트들의 진리표를 나타내었다.

표 1. 새로운 논리값 설명
Table 1. Description of new logic values.

구 분	논리값 설명
기본값	0: 정적 논리값 0 1: 정적 논리값 1 R(t): 0→1로 변하는 논리값. t는 트랜지션 시간. F(t): 1→0로 변하는 논리값. t는 트랜지션 시간. G0: 1→0→1로 변하는 잡음신호 G1: 0→1→0로 변하는 잡음신호 X: 알 수 없는 값

논리값 0과 1은 두 번에 걸쳐 연속적으로 인가되는 정적인(변화하지 않는) 논리값을 표현한다. 즉, 0→0 또는 1→1을 의미한다. 논리값 R, F의 조합은 게이트의 출력단에 잡음 신호를 만들 수 있는데, 논리값 G0 또는 G1을 이용해 이 잡음 신호를 표현한다. Crosstalk로 인한 고장영역(fault site) 이전에 발생한 잡음 신호는 고장영역에까지 전파되지 않는 한 무시할수 있다. 그러나, 고장영역에서 발생한 crosstalk 잡음 신호는 논리값 G0, G1을 이용해 주출력단으로 전파시킨다.

2. 테스트 생성 알고리즘

본 논문에서 제안하는 알고리즘은 PODEM을 기반으로 하고 있으며, 크게 D를 설정하는 부분과 D를 전파시키는 부분으로 구분할 수 있다. 이 알고리즘은 다음 세가지 절차에 따라 수행된다. 첫 번째, 피해선로에 정적인 논리값 0 또는 1을 인가한 후, 두 번째, 능동선로에 논리값 변화(트랜지션)를 발생시킨다. 마지막 세 번째는 발생된 crosstalk 잡음을 주출력단으로 전파시키는 일이다. 제안하는 알고리즘의 개요를 아래에 기술하였으며, 그림 5는 그것을 순서도로 옮겨 놓은 것이다. 본 알고리즘은 모든 crosstalk 고장에 대해 레이블 [P1]에서 시작하여 레이블 [DETECTED] 또는

XOR	0	1	R(t2)	F(t2)	X
0	0	1	R(t2)	F(t2)	X
1	1	0	F(t2)	R(t2)	X
R(t1)	R(t1)	F(t1)	0, t1=t2 G1, t1≠t2	1, t1=t2 G0, t1≠t2	X
F(t1)	F(t1)	R(t1)	1, t1=t2 G0, t1≠t2	0, t1=t2 G1, t1≠t2	X
X	0	X	X	X	X

AND	0	1	R(t2)	F(t2)	X
0	0	0	0	0	0
1	0	1	R(t2)	F(t2)	X
R(t1)	0	R(t1)	R(t2), t1<t2	G1, t1<t2 0, t1>t2	X
F(t1)	0	F(t1)	G1, t1>t2 0, t1<t2	F(t1), t1<t2 F(t2), t1>t2	X
X	0	X	X	X	X

OR	0	1	R(t2)	F(t2)	X
0	0	1	R(t2)	F(t2)	X
1	1	1	1	1	1
R(t1)	R(t1)	1	R(t1), t1<t2 R(t2), t1>t2	1, t1≤t2 G0, t1>t2	X
F(t1)	F(t1)	1	1, t1≥t2 G0, t1<t2	F(t1), t1>t2 F(t2), t1<t2	X
X	0	1	X	X	X

그림 4. 기본 게이트에 대한 진리표
Fig. 4. Truth tables of basic gates.

[UNDETECTED]에서 종료한다.

1. 모든 능동선로에 논리값 R, F가 인가되었으며 인가된 논리값이 미리 지정된 오차범위 내에서 동일한 타이밍인지 검사한다. 이 조건이 만족된다면 레이블 [P2]로 진행한다. 그러나 조건이 만족되지 않는다면 능동선로는 초기 목표선로로, 논리 자극값인 R, F는 초기 목표값으로 사용되며, 제어는 레이블 [SIO](Select Initial Objectives)로 옮겨간다. 만일 R 또는 F를 능동선로에 인가할 수 없다면 제어는 [BT](Back Track)로 이동한다.
2. 피해선로에 능동선로의 논리값 R, F에 대응하는 논리값 0, 1이 인가되었는지 검사한다. 또한 타이밍은 미리 지정된 오차범위 내에서 능동선로의 논리값 변화보다 빠르거나 같아야 한다. 이 조건을 만족한다면 제어를 레이블 [P3]로 이동하고, 조건을 만족하지 않는다면 피해선로는 초기 목표선로로, 피해선로의 논리값 0, 1은 초기 목표값이 되어 레이블 [SIO]로 제어를 이동한다. 피해선로에 고장 논리값 G0, G1을 발생시킬 수 없다면 제어는 [BT](Back Track)로 이동한다.
3. 잡음신호 G0 또는 G1이 회로의 주 출력단 중 어느 한 곳으로 전파되는지 검사한다. 잡음이 전파되었다면 제어는 [DETECTED]로 이동하며, 전파되지 않았다면 또 다른 주 출력단을 초기 목표선로로 지정한 후 레이블 [SIO](Select Initial Objectives)로 제어를 이동한다. 그러나, 잡음신호를 전파할 주 출력단이 존재하지 않을 경우 제어는 [BT]로 이동한다.
4. 초기 목표선로로부터 주입력단에 도달할 때 까지 후방추적(back trace)을 행하여 입력신호가 X인 게이트를 선택한다. 주입력 게이트가 선택되면 해당 게이트의 입력값과 그에 따른 출력 논리값을 결정하고 제어를 [P5]로 이동한다.
5. 새로이 결정된 주입력 단의 논리값에 대한 전방영향분석(forward implication)을 수행한다. 이 과정의 결과가 능동선로와 피해선로의 초기목표값이 아닐 때 실패로 간주되며 제어는 [BT]로 이동한다.
6. 시도철회(back track)를 거쳐 위 5번째 과정(레이블 [P5])이 반복된다. 모든 주입력단에서 논리값이 결정되지 못하면 테스트 생성은 실패하고 제어는 [UNDETECTED]로 이동한다.
7. 피해선로로부터 고장영향이 전파된 주출력단(Primary output)을 관찰 터미널(observation terminal)로 하며, 이 때 주입력단의 입력값들을 테스트 패턴으로 한다.
8. 더 이상의 목표고장이 없다면 테스트 생성은 종료된다.

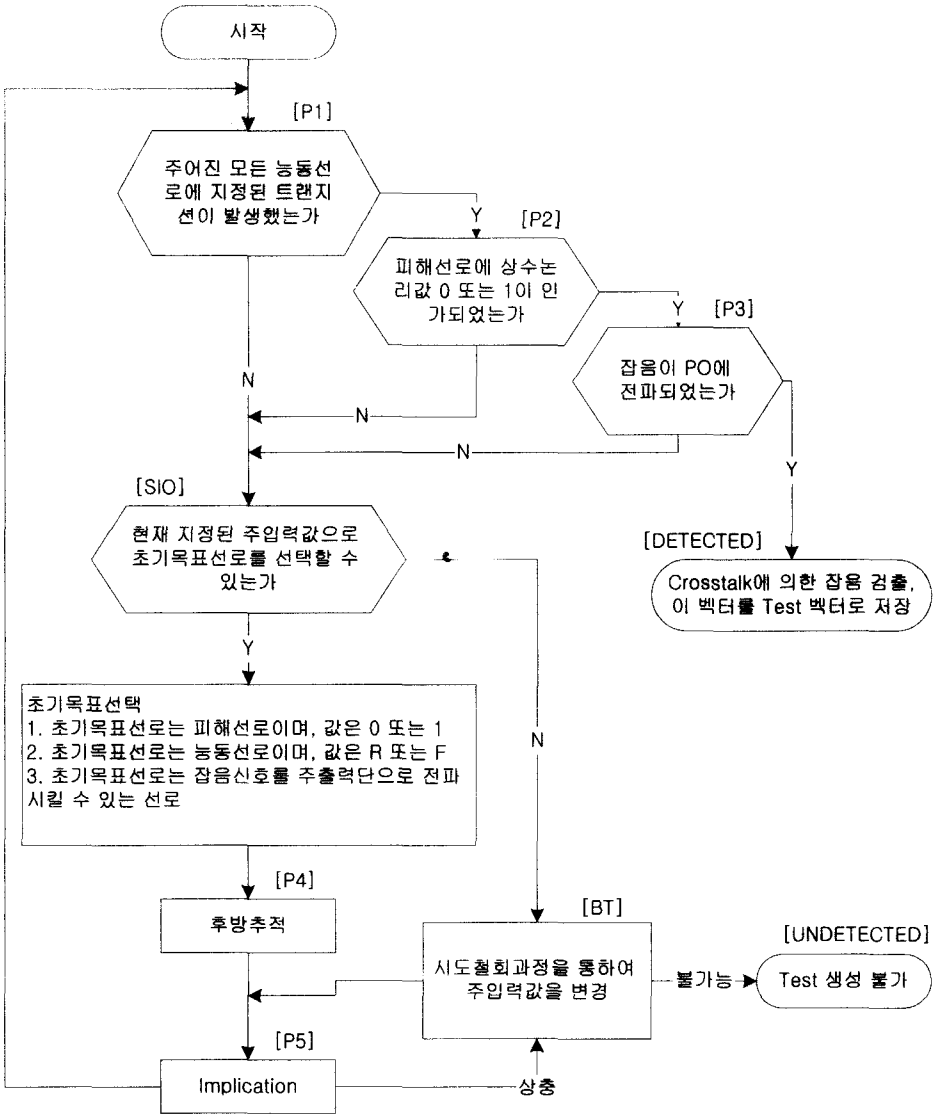


그림 5. 제안하는 알고리즘
Fig. 5. Flowchart of proposed algorithm.

3. 타이밍 조건

피해선로에 충분히 안정된 논리값 0 또는 1이 흐르고 있을 때 능동선로의 신호 변화는 피해선로에 crosstalk 잡음을 일으킨다. 만일 피해선로의 신호가 안정되는 시간에 비해 능동선로의 신호 변화가 빠른 경우, 피해선로에 약한 잡음이 발생하여 직후 게이트까지 잡음신호가 전달되지 않거나 아예 잡음이 발생하지 않을 수도 있다. 그러므로 피해선로는 능동선로의 신호 변화보다 빠르거나 같은 시간에 안정된 신호를 유지하고 있어야 한다. 또한, 피해선로에 잡음 효과를 극대화하기 위해 모든 능동선로에서 일어나는 신호변화는 동일한 시간이어야 할 필요성이 있다. 그러나 이는 현실적으로 매우 불가능한 경우이므로 어느 정도의 시간 오차를 허용한다. 오차 범위는 VLSI 설계에 따라 다양할 수 있으므로 본 논문에서는 시간 오차를 프로그램 수행시 환경설정을 통해 인위적으로 지정한 후 실험을 수행하였다.

피해선로에 crosstalk 잡음고장을 발생시키기 위한 논리값 정당화 과정의 타이밍 조건은 다음과 같다.

$T_{affecting_i} \geq T_{victim}$, $i < n$, n은 모든 능동선로의 총 수

여기서, 능동선로 $T_{affecting_i}$는 오차범위 r에 대해,

$$T_{affecting_i} - r \leq T_{affecting_i} \leq T_{affecting_i} + r$$

이다.

Ⅳ. 실험 결과

본 논문에서 제안하는 테스트 생성 알고리즘은 C 언어를 통해 구현하여 ISCAS '85 벤치마크 회로에 적용하였으며, 프로그램은 512MB의 메인메모리를 갖는 Sun Sparc 5 머신에서 실행하였다. 주어진 벤치마크 회로가 가지는 crosstalk 고장 정보, 회로의 레이아웃, 그리고 타이밍에 관련된 정보는 알 수 없기 때문에 고장목록을 만들어 주는 별도의 프로그램을 작성하여 무작위로 crosstalk 고장 위치를 지정하였으며 고장의 수는 회로당 500개를 가정하였다. 타이밍 지정에는 게이트 넷-리스트의 1 레벨을 1 단위시간(unit time)으로 하는 단위-타이밍 지정 방법(unit timing method)을 사용하였다.

능동선로의 수가 여러 가지로 변화하는 경우의 테스트 생성 결과를 표 2에 표시하였다. ISCAS '85 벤치마크 회로 중 무작위로 하나를 선정하여 능동선로의 수를 3개에서 6개까지 늘려가며 테스트를 생성한 결과이다. 알고리즘을 레이블 [DETECTED]에서 끝낸 횟수를 Detected 항목에 기재하였고, 테스트 생성이 불가하여 레이블 [UNDETECTED]에서 알고리즘을 끝낸 경우엔 ①시도철회 제한 횟수를 초과하여 테스트 생성이 중단된 경우 Aborted 항목에, ②테스트 패턴을 찾을 수 없어 알고리즘을 탈출한 경우를 Undetected 항목으로 구분하여 기술하였다. TG Rate(Test Generation Rate) 항목은 테스트 생성의 효율성을 평가하기 위한 수치로서 참조논문 [7]의 방법을 이용했다. 이와는 달리, 여러 가지 ISCAS '85 벤치마크 회로를 대상으로 2개의 능동선로만을 가정한 경우를 표 2와 동일한 형식으로 표 3에 표시하였다.

표 2. 능동선로의 수에 따른 테스트 생성 결과

Table 2. Experiment results according to number of affecting lines.

Affecting line	Detected	Undetected	Aborted	TG Rate	Time (s)
3	282	177	41	91.8%	32
4	243	223	34	93.2%	38
5	198	254	48	90.4%	42
6	188	257	55	89.0%	44

표 3. ISCAS'85 벤치마크 회로에 대한 테스트 생성 결과

Table 3. Experiment results for ISCAS '85 Benchmark circuits.

Circuit	Detected	Undetected	Aborted	TG Rate	Time(s)
c1355	166	322	12	97.6%	21
c1908	307	163	30	94.0%	15
c2670	359	82	59	88.2%	19
c3540	106	298	96	80.8%	59
c5315	339	121	40	92.0%	24
c6288	155	311	34	93.2%	381
c7552	304	174	22	95.6%	36

Ⅴ. 결론 및 향후 연구 계획

본 논문에서는 논리 수준에서 crosstalk 잡음 고장을 검출하는 ATPG 알고리즘과 새로운 모델링 방법을 제시하였고, 실제로 구현하여 테스트 생성 결과를 출력하였다. 버스라인과 같이 원거리에 신호를 전달하는 다중 선로 상에서 crosstalk 잡음이 발생할 확률이 높다는 사실에 착안함으로써 다중 선로의 경우에 훨씬 유용함에 중점을 두고 접근하였다. 알고리즘은 전통적인 PODEM 알고리즘을 바탕으로 피해선로 및 다수의 능동선로를 포함하는 여러 초기목표를 동시에 달성할 수 있도록 개선하였다.

실험 결과는 현저히 낮은 테스트 시간비용에 비해 높은 테스트 생성 효율을 보여주고 있으나, 높은 테스트 생성 효율에 비해 실제 고장 검출률은 그리 높지 않음을 알 수 있다. 테스트 생성 시간 면에서 참조논문 [7]에 비해 적게는 57배, 많게는 150배에 이르는 개선이 이루어졌으나, 고장검출이 가능한 테스트패턴을 생성한 경우, 즉 레이블 [Detected]에서 알고리즘이 종료된 비

율은 참조논문 [7]과 비교해 -13.8% ~ 20% 정도의 개선이 이루어졌다. 여러 개의 초기목표를 동시에 만족시켜야 하는 본 논문의 모델링 특성은 회로의 각 모듈에 내장된 팬-아웃 선로로 인해 테스트 생성이 불가한 경우를 상당수 만들어 내고 있는 것으로 생각된다. 앞으로의 연구를 통하여 이러한 문제점을 해결하고자 한다.

참 고 문 헌

[1] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Analytic models for crosstalk delay and pulse analysis for non ideal inputs", *Proceedings of International Test Conference*, pp. 809~818, 1997.

[2] A. Vittal and M. Marek-Sadowska, "Crosstalk reduction for VLSI", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 16, pp. 290~298, March 1997.

[3] K. Rahmat, J. Neves, J. Lee, "Methods for calculating coupling noise in early design: a comparative analysis", *Proceedings of International Conference on Computer Design VLSI in Computers and Processors*, pp. 76~81, 1998.

[4] H. Kawaguchi and T. Sakurai, "Delay and Noise Formulas for Capacitively Coupled Distributed RC Lines", *Proceedings of the Asian and South Pacific Design Automation Conference*, pp. 35~43, 1998.

[5] A. Rubio, N. Itazaki, X. Xu, and K. Kinoshita, "An approach to the analysis and detection of crosstalk faults in digital VLSI circuits", *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 13, pp. 387~394, March 1994.

[6] K. T. Lee, C. Nordquist, and J. Abraham, "Automatic Test Pattern Generation for Crosstalk Glitches in Digital Circuits", *Proceedings of IEEE VLSI Test Symposium*, pp. 34~39, 1998.

[7] W. Y. Chen, S. K. Gupta, and M. A. Breuer, "Test Generation in VLSI Circuits for Crosstalk Noise", *Proceedings of International Test Conference*, pp. 641~650, 1998.

[8] 김석윤, VLSI 시스템 회로연결선의 모형화 및 해석, IDEC 교재 개발 시리즈 10, 시그마 프레스, 1999

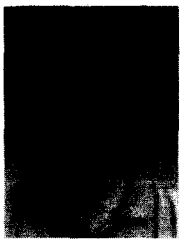
[9] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits", *IEEE Transactions on Computers*, Vol. C-30, March, 1981.

[10] K. T. Lee, C. Nordquist, and J. Abraham, "Test Generation for Crosstalk Effects in VLSI Circuits", *IEEE International Symposium on Circuits and Systems*, Vol. 4, pp. 628~631, 1996.

저 자 소 개

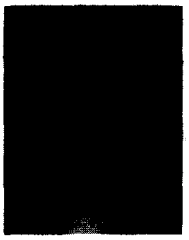
李 永 均(正會員)

1999년 숭실대학교 컴퓨터학부 학사. 2002년 숭실대학교 대학원 컴퓨터학과 석사. 2002년~숭실대학교 대학원 박사과정. <주관심분야 : 컴퓨터구조, VLSI 설계 및 테스팅, CAD>



梁 善 雄(正會員)

1999년 숭실대학교 전자계산학과 학사. 1998년 숭실대학교 대학원 전자계산학과 석사. 2002년 숭실대학교 대학원 박사. <주관심분야 : 컴퓨터구조, VLSI 설계 및 테스팅, CAD>



金 文 俊(正會員)

2000년 숭실대학교 컴퓨터학부 학사. 2002년 숭실대학교 대학원 컴퓨터학과 석사. 2002년~숭실대학교 대학원 박사과정. <주관심분야 : 컴퓨터구조, VLSI 설계 및 테스팅, CAD>

張 勳(正會員)

1987년 서울대학교 전자공학과 학사. 1989년 서울대학교 전자공학과 석사. 1993년 University of Texas at Austin 박사. 1991년 IBM Inc. 1993년 Motorola Inc. Senior Member of Technical Staff. 1994년~현재 숭실대학교 컴퓨터학부 교수. <주관심분야 : 컴퓨터 시스템, VLSI 설계, VLSI 테스팅>