한국산학기술학회논문지
Vol. 4, No. 1, pp. 7-12, 2003

신호선의 상관관계를 고려한 개선된 테스트용이도 분석 알고리즘

김 윤 홍*

An Improvement on Testability Analysis by Considering Signal Correlation

Yun Hong Kim*

요 약 테스트용이도(testability) 분석은 논리회로에서 발생하는 stuck-at 고장을 테스트하는 것이 어느 정도 어려운가를 예측 평가하기 위한 목적에서 이루어진다. 좋은 테스트용이도 분석 프로그램이 있다면, 회로의 테스트용이도를 개선하기 위한 좋은 방안을 회로 설계자들에게 사전에 제시해줌으로써, 테스트 문제에 미리 대비할 수 있도록 해준다. 그 동안 테스트용이도 분석을 효율적으로 수행하기 위한 연구가 있었다. 그러나 COP이나 SCOAP과 같은 기존의 대표적인 테스트용이도 분석 알고리즘들은 트리 구조를 갖는 회로의 경우에 각 stuck-at 고장의 테스트용이도 값을 효율적으로 계산할 수 있으나, 일반적인 구조의 회로에 대해서는 정확도가 떨어진다. 그 이유는 테스트용이도 분석을 선형적인 시간 내에 수행하기 위해서, 각 신호선들은 재수렴 팬아웃(reconvergent fanout)으로 인한 상관관계가 없는 것으로 가정하기 때문이다. 본 논문에서는 테스트용이도 분석을 위해 신호선 상관관계를 고려한 개선된 방법을 제안한다. 제안된 방법에서는, 회로 내에서 재수렴 팬아웃과 이에 영향을 받는 게이트들에 대한 정보를 사전에 파악하기 위한 재수렴 팬아웃 분석 알고리즘을 이용하여, 재수렴 팬아웃으로 인한 효과를 테스트용이도 분석에 반영함으로써 정확도를 높이고 있다.

Abstract The purpose of testability analysis is to estimate the difficulty of testing a stuck-at fault in logic circuits. A good testability measurement can give an early warning about the testing problem so as to provide guidance in improving the testability of a circuit. There have been researches attempting to efficiently compute the testability analysis. Conventional testability measurements, such as COP and SCOAP, can calculate the testability value of a stuck-at fault efficiently in a tree-structured circuit but may be very inaccurate for a general circuit. The inaccuracy is due to the ignorance of signal correlations for making the testability analysis linear to a circuit size. This paper proposes an efficient method for computing testability analysis, which takes into account signal correlation to obtain more accurate testability. The proposed method includes the algorithm for identifying all reconvergent fanouts in a given circuit and the gates reachable from them, by which information related to signal correlation is gathered.

Key Words : SCOAP, signal correlation, testability

1. 서 론

테스트용이도(testability)는 반도체회로의 설계 및 생산이 관련된 여러 분야의 시각차에 따라 약간은 다르게 정의될 수 있으나, 공통적으로는 '테스트를 쉽게 만드는 회로의 성질'로 정의될 수 있다. 테스트용이도 분석의 가장 기본적인 목적은 회로를 설계하면서 앞으로 발생할 테스트 문제에 대한 사전 정보를 제공하는 것이다. 테스트용이도 분석은 회로 구조를 토대로 하여 이루어지며, 회로의 각 신호선들에 대한 제어도(controllability) 및 관측도(observability)와 같은 파라미터들을 결정한다. 이 정보들은 테스트가 용이하도록 회로를 설계해 나가는데 유용하게 사용된다[1-3]. 회로의 테스트용이도를 평가하기 위해 사용되는 또 다른 방법은 고장시뮬레이션(fault simulation)이다. 이것은 주어진 테스트 패턴이 회로의 신호선들을 테스트할 때 어느 정도의 효과가 있는지를 평가하는데 사용된다. 고장 시뮬레이션에 비하여 테스트용이도 분석은 특정 테스트 패턴 집합에 의존하지 않기 때문에 더욱 경제적이고, 그 결과 또한 더 폭넓게 적용가능하다. 테스트용이도가 유용하게 사용되려면 테스트패턴 생성이나 고장시뮬레이션보다 단

*상명대학교 천안캠퍼스 공과대학 컴퓨터 · 정보 · 통신 공학부 컴퓨터시스템공학 전공
Tel. 041-550-5358, e-mail: yhkim@smuc.ac.kr
"본 논문은 상명대학교 2001학년도 교내연구비 지원에 의하여 연구되었음"

순해야 하는데, 이를 위해서 테스트용이도 분석은 보통 2가지 특성을 갖는다. 첫째는 회로의 구조만을 분석한다는 것이다. 따라서 어떤 테스트 패턴과도 관련되지 않는다. 이는 회로설계의 초기 단계에서부터 테스트용이도 분석의 사용이 가능하다는 것을 의미한다. 두 번째 특성으로 테스트용이도 분석 알고리즘은 회로크기에 선형적이어야 한다는 것이다. 그렇지 않으면 더 정확한 데이터를 제공해주는 테스트패턴 생성이나 고장시뮬레이션을 사용할 수도 있기 때문이다.

테스트용이도 분석 분야에서 초기의 연구 노력은 회로크기에 선형적인 복잡도를 갖는 분석방법에 대해 주로 이루어졌다[4-9]. 그러나 이런 방법들의 정확도는 만족할 수준에 못미치는 것이었다. 통상적으로 테스트용이도 분석은 회로의 각 신호선에 대한 제어도와 관측도를 계산함으로써 각 신호선의 테스트용이도 값을 결정한다. 계산방법은 각 분석알고리즘이 제시하는 규칙에 따라 입력에서 출력 방향으로 제어도를 계산한 후, 다시 출력에서 입력 방향으로 관측도를 계산함으로써 두 번의 처리를 거쳐 이루어진다. 테스트용이도는 계산절차를 단순화하기 위해 몇 가지 제한적 가정을 하기 때문에 그 결과값은 근사치가 되며, 따라서 테스트용이도 분석은 테스트패턴 생성이나 고장시뮬레이션보다 시간은 적게 소모되지만 결과데이터의 정확도는 떨어진다.

테스트용이도 분석도구인 SCOAP[4]은 각 신호선에 대한 테스트의 어려움을 비용 개념으로 표현한 수치를 이용한다. SCOAP에서 각 신호선이 갖을 수 있는 값의 범위는 0에서 ∞이다. 이 값들은 각 신호선을 제어하거나 관측하는데 요구되는 노력을 수치화한 것으로, 값이 클수록 더 큰 노력이 요구됨을 뜻한다. SCOAP은 계산을 단순화하기 위해 신호선 간의 상관관계는 없는 것으로 가정한다. COP[5] 알고리즘은 비용의 개념이 아닌 확률적인 접근방법으로 각 신호선의 제어도와 관측도를 표현하였다. 그러나 재수렴 팬아웃으로 인한 신호선 간의 상관관계를 무시함으로써, COP에 의해 계산된 확률값은 일반적으로 부정확하다. PREDICT[10] 알고리즘은 super gate라는 개념을 이용하여 신호선 확률값을 개선하려고 하였다. 이에 따라 PREDICT는 비교적 정확한 테스트용이도를 계산하지만 상당한 계산복잡도가 요구된다.

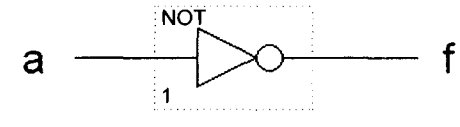
본 논문에서는 신호선 상관관계를 고려한 개선된 테스트용이도 분석 알고리즘을 제안한다. 제안된 테스트용이도 분석 알고리즘은 재수렴 팬아웃 정보를 분석하여 저장한 후에 이를 SCOAP에서 반영하여 테스트용이도를 계산하도록 처리함으로써, 기존의 SCOAP보다 정확한 테스트용이도를 계산할 수 있다. 또한 재수렴 팬아웃에 대한 별도의 사전 처리를 함으로써, 재수렴 팬아웃으로 인해 제어가 불가능한 신호선을 미리 파악하여 제어도를 결정하여 테스트용이도의 정확도를 높인다.

본 논문의 구성은, 우선 2절에서 테스트용이도 분석의 기본 개념 및 문제점에 대하여 논의하고, 3절에서는 테스트용이도 분석 알고리즘을 제시하고, 제시된 알고리즘에 대한 검토를 하며, 마지막으로 4절에서 결론을 내린다.

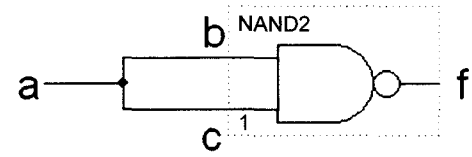
2. 테스트용이도 분석의 기본 개념 및 문제점 논의

SCOAP은 회로의 각 신호선에 대한 0의 제어도(CY0)와 1의 제어도(CY1)를 구분하여 사용하며, 관측도는 구분없이 사용한다. 이 값들은 요구되는 테스트 비용이 증가할수록 커지게 되어, 회로구조만을 고려하여 내부 신호선의 논리값을 제어하고 관측하는 어려움에 대한 정량적인 평가치를 제공한다. SCOAP은 각 게이트의 관점에서 입력과 출력의 논리적인 관계만을 고려하여 테스트용이도의 계산이 진행되기 때문에 회로 전체적인 관점에서의 관계를 반영하지 못하고 있다. 예를 들어, 그림 1은 인버터를 두 가지 방법으로 구현한 논리회로를 보여준다.

이 두 회로는 동일한 논리동작을 하는 회로이지만, SCOAP은 서로 다른 제어도(CY0)를 계산해 낸다. 그 이유는 그림 1(b)의 NAND 구현에서 신호선 b와 c가 사실은 동일한 신호선 a에 연결된 것이지만, SCOAP은 b와 c가 서로 독립적인 신호선인 것으로 가정하고 제어도를 계산하기 때문이다. 즉, b를 1로 만드는 비용과 c를 1로 만드는 비용은 별도의 비용인 것으로 SCOAP은 처리하지만, 사실은 a를 1로 만드는 비용으로 모두 만족시킬 수 있다. b의 제어도 CY1(b)과 c의 제어도 CY1(c)에는 이미 CY1(a)의 값이 공통적으로 포함되어 있는 것이다. 따라서 CY0(f)는 CY1(a)값이 중복 계산



(a) 인버터 구현



(b) NAND 구현

그림 1. 인버터의 두 가지 구현.

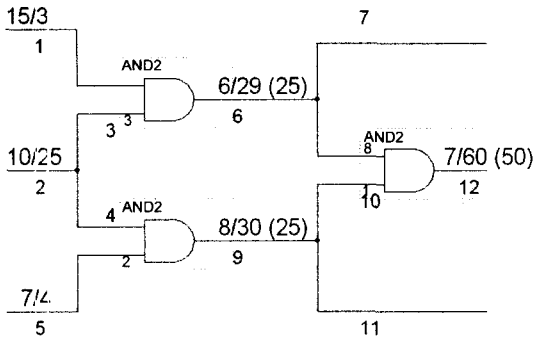


그림 2. 예제 회로 1.

되어 결과값이 부정확하게 된다. 이런 경우처럼 게이트의 입력선을 무조건 독립적인 것으로 가정하고 계산을 간단화하는 것은 결과의 정확도를 떨어뜨린다. SCOAP 뿐만 아니라 대부분의 테스트용이도 분석 알고리즘들이 이와 유사한 가정을 하고 있기 때문에, 특히 재수렴 팬아웃이 많은 회로의 경우에 테스트용이도 값은 그 정확도가 상당히 떨어진다. 물론 그런 가정으로 인해 정보가 왜곡되는 정도는 회로설계에 따라 그리고 그 값이 해석되는 방식에 따라 변화 폭이 유동적이다.

그림 1의 경우보다 한 단계 확장된 경우로서 그림 2의 예제회로 1을 살펴보자. 예제 회로1에서 각 신호선의 CY0과 CY1은 CY0/CY1로 표기한다. ( )안의 값은 신호선 2의 재수렴 팬아웃으로 인하여 다른 신호선의 CY1값에 포함된 CY1(2)값을 보여준다.

그림 2의 회로에서는 신호선 2가 팬아웃되어 서로 다른 경로를 거친 후에 AND게이트를 통하여 신호선 12로 재수렴하고 있다. 신호선 12의 1에 대한 제어도는 CY1(12)=CY1(8)+CY1(10)+1=29+30+1=60에 의해 계산된다. 그러나 CY1(8)=29에는 CY1(2)값인 25가 포함된 값이며, CY1(10)=30에도 CY1(2)의 값이 포함되어 있다. 중복 계산된 것을 감안하면 CY1(12)=(29+30+1)-25=35와 같이 계산되어야 한다. SCOAP에 의해 계산된 값과 수정계산된 값과는 25의 큰 차이가 발생하며, 이런 값의 차이로 인해 이후의 테스트용이도 계산이 영향을 받아 모두 부정확하게 된다. 따라서 SCOAP에 의해 계산된 테스트용이도 값을 정확히 수정하기 위해서는, 입력신호선들이 동일한 재수렴 팬아웃의 영향을 받는지 여부에 관한 정보를 사전에 분석 저장한 후 테스트용이도를 계산하는 동안에 이 정보를 이용하여 정확한 값으로 수정해 나가야 한다.

그림 3의 또 다른 예를 살펴보자.

그림 3의 회로에서 SCOAP에 의해 계산된 신호선 5의 CY0과 CY1은 각각 2와 4이다. 그러나 회로구조를 살펴보면, 신호선 1에 어떠한 값이 놓여도 신호선 5는 항상 0값을 갖으며 1값이 될 수는 없다. 즉, 실제는 CY0(5)=0이고 CY(1)=∞가 된다. SCOAP을 포함한 대부분의 테스트용이도 분석 알고리즘은 각 게이트만을 보면서 테스트용이도 값을 계산해 나가기 때문에, 이와 같은 회로구조에 대해 적절히 대처하지 못하고, 실질적인 값과는 상당한 차이가 난다. 결국 4와 ∞라는 값의 큰 차이로 인해 나머지 테스트용이도 계산의 정확도에 상당한 영향을 준다.

바람직한 테스트용이도 분석 알고리즘이라면 제어나 관측이 불가능한 신호선에 대해 정확한 테스트용이도 값을 할당할 수 있어야 한다. 테스트용이도 분석 알고리즘에서 이 문제를 효과적으로 해결하기 위해서는 재수렴 팬아웃으로 인해 제어 불가능한 특정 신호선을 미리 파악하여 정확한 테스트용이도 값을 미리 할당하는 과정이 필요하다.

재수렴 팬아웃은 제어도에서와 마찬가지로 관측도의 계산에도 많은 영향을 준다. 그림4의 간단한 예제회로를 보자.

관측도는 제어도 계산이 모두 끝난 후에 주출력에서 주입력 방향으로 계산이 진행된다. 그림 4에서 신호선 a에 대한 관측도는 SCOAP의 관측도 계산방식에 따라 OY(a)=OY(f)+CY0(b)+CY0(c)+1로 계산된다. 그러나 신호선 b와 c는 동일한 재수렴 팬아웃에 의해 영향을 받고 있기 때문에, CY0(b)와 CY0(c)에는 재수렴 팬아웃의 제어도가 모두 포함되어 있어서 OY(a)의 값은 그 중복된 값만큼 부정확하게 된다.

이상에서 살펴본 바와 같이 SCOAP과 같은 테스트용이도의 정확도를 높이기 위해서는 재수렴 팬아웃으로 인한 신호선 간의 상관관계를 미리 파악하여 이를 테스트용이도 계산 알고리즘에 반영하는 것이 필요하며, 이를 반영한 테스트용이도 계산 알고리즘이 다음 절에서

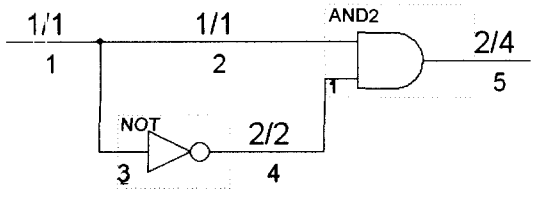


그림 3. 예제 회로 2.

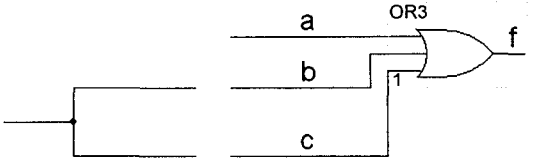


그림 4. 예제 회로 3.

제안된다.

3. 테스트용이도 분석 알고리즘

2절에서 언급한 것처럼, SCOAP과 같은 테스트용이도 분석 알고리즘들이 게이트에 한정된 입출력 관계만을 고려하여 계산하는 방식을 개선하려면, 회로 구조의 전체적인 관점에서 그 관계를 고려하면서 테스트용이도를 계산할 수 있어야 한다. 특히 신호선들이 서로 독립적이지 못하고 상관관계를 갖게 되는 주된 이유가 재수렴 팬아웃이기 때문에, 회로 내에서 재수렴 팬아웃과 이에 영향을 받는 게이트들에 대한 정보를 사전에 파악할 수 있어야 한다.

주어진 회로 내에서 재수렴 팬아웃 정보를 추출하기 위한 재수렴 팬아웃 분석 알고리즘은 알고리즘 1과 같다. 팬아웃 스템은 전처리과정에서 게이트 레벨에 따라 일렬번호가 붙여진다.

회로에서 팬아웃 스템(fanout stem)은 여러 개의 팬아웃 가지(fanout branch)로 갈라지는데, 팬아웃 가지들이 여러 경로를 통하여 다시 동일한 게이트에서 만나게 되면 그 팬아웃 스템은 재수렴 팬아웃이 된다. 재수렴 팬아웃 분석 알고리즘은 회로가 주어지면, 모든 팬아웃 스템을 주입력부터 주출력까지 게이트 레벨의 증가 순으로 하나씩 선택하여 해당 팬아웃 스템으로부터 도달할 수 있는 모든 신호선들에는 해당 팬아웃 스템 번호를 기록하게 된다. 모든 팬아웃 스템에 대한 처리가 끝나면, 각 게이트의 입력신호선에 기록된 팬아웃 스템 번호를 비교하여 공통된 팬아웃스템 번호가 있는가, 그리고 있다면 공통된 팬아웃 스템 번호 중에서 가장 큰 번호가 무엇인지를 파악한다. 공통된 팬아웃 스템 번호 중에서 낮은 번호의 팬아웃 스템은 실제로 재수렴되지 않는다. 그 이유는 가장 큰 번호의 팬아웃 스템으로 인하여 재수렴되는 것처럼 보일 뿐이다.

재수렴 팬아웃 분석과정을 포함한 전체적인 테스트용이도 분석 알고리즘은 알고리즘 2와 같다.

```
void  IdentifyAllReconvergentFanouts(circuit C)
{
    for (회로 C 에서 게이트 레벨의 오름차순에 따른 각 팬아웃 FSi) {
        FSi 로부터 주출력에 이르는 모든 신호선에 i 를 기록한다;
    }
    for (팬아웃으로부터 도달가능한 각 게이트 Gj) {
        if (2(동일한 팬아웃 번호를 갖는 서로 다른 Gj 입력이 존재)
            continue;
        num = Maximum(Gj 의 서로 다른 입력들 중에서 공통 팬아웃 번호들);
        FSnum 과 Gj 를 재수렴 팬아웃 목록에 기록한다;
    }
}
```

알고리즘 1. 재수렴 팬아웃 분석 알고리즘.

```
void  TestabilityAnalysis(circuit C)
{
    IdentifyAllReconvergentFanouts(C);
    InitializeTestability(C);
    for (게이트 레벨의 오름차순에 따른 각 게이트 Gi) {
        if (CY(Gi)가 이미 설정)
            continue;
        SCOAP 에 따라 CYSCOAP(Gi)을 계산한다;
        if (Gi 가 재수렴 게이트) {
            for (Gi 에서 수렴하는 각 재수렴 팬아웃스템 RFSj) {
                n = RFSj 의 영향을 받는 Gi 입력수;
                CY(Gi) = CYSCOAP(Gi) - (n - 1) × CY(RFSj);
            }
        } else {
            CY(Gi) = CYSCOAP(Gi);
        }
    }
    게이트 레벨의 내림차순으로 SCOAP 에 따라 OYSCOAP 을 먼저
    계산한 후, 재수렴 팬아웃의 영향을 고려하여 제어도 계산절차와
    동일하게 관측도를 수정한다;
}
void InitializeTestability(circuit C)
{
    CY0(모든 PI) = CY1(모든 PI) = 1;
    OY(모든 PO) = 1;
    for (각 재수렴 팬아웃스템 RFSi) {
        RFSi 에 0 과 1 을 각각 할당하고 implication 을 수행한다;
        switch (재수렴 게이트의 출력값) {
            case 항상 0:
                CY0 = 0; CY1 = ∞; break;
            case 항상 1:
                CY0 = ∞; CY1 = 0; break;
        }
    }
}
```

알고리즘 2. 테스트용이도 분석 알고리즘.

우선 회로가 주어지면 재수렴 팬아웃을 파악한 후에, InitializeTestability( )에서 각 신호선의 제어도와 관측도를 초기화한다. InitializeTestability( )에서는 SCOAP에 의한 기본적인 초기화 뿐만 아니라, 재수렴 팬아웃 스템에 각각 0과 1값을 할당한 후 이에 대한 implication을 수행한다. 이로부터 재수렴 게이트 출력값이 변하는지 여부를 확인하여 재수렴 게이트 출력이 항상 일정한 0(1)값을 갖을 경우에는 1(0)로 제어가 불가능하므로 CY1(CY0)은 ∞가 되고, 반대로CY0(CY1)은 0이 된다. 이와 같이 제어가 불가능한 신호선을 미리 파악하여 해당 신호선의 제어도를 미리 결정한다. 따라서 제안된 테스트용이도 분석 알고리즘은 SCOAP과 달리, 제어가 불가능한 신호선에 대해 ∞값을 할당할 수 있음으로써 해당 신호선에 대한 정확한 정보를 제공하며, 또한 이 신호선의 영향을 받는 이후의 신호선들의 제어도 계산에도 정확성을 높일 수 있다.

InitializeTestability( )가 끝나면 주입력에서 주출력

방향으로 각 게이트에 대한 제어도를 SCOAP에 따라 계산해 나간다. 게이트의 제어도가 이미 초기화과정 중에 계산되었을 경우에는 건너뛴다. 재수렴 게이트일 경우에는 이에 영향을 주는 재수렴 팬아웃들을 목록에서 파악한 후, 각 재수렴 팬아웃이 영향을 미치는 재수렴 게이트 입력선 수 n을 계산한다. 결국 재수렴 게이트의 최종 제어도값은 $CY_{SCOAP}(G) - \sum_{j=1}^{m} (n_j - 1) \times CY(RFS_j)$ 이 된다. 여기서 m은 재수렴 게이트에 영향을 주는 재수렴 팬아웃의 개수이다. 수식에서 일관된 표기에 어려움이 있어 CY로 간단히 표현하였지만, 실제로 게이트 타입(AND, OR, NAND, NOR 등)에 따라 적절히 CY0 또는 CY1로 바뀌어야 한다.

제어도 계산이 완료되면, 주출력에서 주입력 방향으로 관측도를 계산한다. 일반 게이트인 경우에는 SCOAP에 따라 관측도를 계산하지만, 재수렴 게이트일 경우에는 현재 관측도를 계산하고 있는 입력이 아닌 나머지 입력들의 제어도에 공통으로 영향을 미치는 재수렴 팬아웃을 파악하여, 중복된 제어도만큼을 빼게 된다. 즉 재수렴 게이트의 한 입력선 a에 대한 관측도는 $OY_{SCOAP}(a) - \sum_{j=1}^{k} (p_j - 1) \times CY(RFS_j)$ 이 된다. 여기서 k는 입력선 a를 제외한 나머지 입력선에 영향을 주는 재수렴 팬아웃의 개수이다. p는 각 재수렴 팬아웃이 영향을 미치는 재수렴 게이트 입력선(a는 제외) 수이다.

이상에서 살펴본 바와 같이 제안된 테스트용이도 분석 알고리즘은 SCOAP과 같은 기존의 테스트용이도 분석 알고리즘에서 고려하지 않고 있는 재수렴 팬아웃으로 인한 신호선의 상관관계를 고려하여 보다 정확한 테스트용이도를 계산한다. 이를 위한 부가적인 처리절차로서 재수렴팬아웃 분석과 이에 따른 테스트용이도 재계산과정이 기존의 테스트용이도 분석 알고리즘에 추가되는데, 이는 기존의 테스트용이도 분석알고리즘의 시간복잡도 O(G)에 큰 영향을 주지 않는다. 여기서 G는 회로의 게이트 수이다.

제안된 테스트용이도 분석 알고리즘의 적용 결과는 회로구조에 따라 다양하게 나타난다. 재수렴 팬아웃이 없는 트리 형태의 회로인 경우에는 기존의 SCOAP과 그 결과가 동일하다. 그 이유는 재수렴 팬아웃이 없기 때문에 신호선 사이의 상관관계가 없고, 따라서 제안된 알고리즘과 SCOAP과의 처리결과에 차이가 없기 때문이다. 한편 재수렴 팬아웃이 많은 회로의 경우에는 대부분 SCOAP 결과와 차이가 비교적 많이 난다. 그러나 재수렴 팬아웃 수가 많은 회로의 경우에도 회로 구조에 따라 그 결과가 SCOAP과 비슷한 경우도 있었다. 이는 재수렴 팬아웃의 영향이 재수렴 게이트에서 비제어값(noncontrolling value)에 나타나지 않고 제어값(controlling value)에 나타나기 때문이다.

4. 결 론

반도체회로의 설계와 제조 분야에서 테스트의 중요성은 날로 높아지고 있으며, 이에 따른 테스트 비용이 반도체회로 제조비용의 커다란 부분을 차지하고 있다. 이러한 추세에 따라, 테스트가 적은 비용으로 짧은 시간 안에 수행될 수 있도록 회로 설계시부터 다양한 testable design기법이 도입되고 있다. 설계된 회로가 어느 정도로 테스트가 쉬운지 그리고 testable design을 도입할 경우 얼마나 개선될지 등을 객관적으로 평가할 수 있는 척도로서 테스트용이도가 사용된다. 그러나 기존의 테스트용이도 분석 알고리즘들은 선형적인 시간내에 분석을 끝내기 위하여 여러가지 제한적 가정을 하고 계산하기 때문에, 테스트용이도 값은 근사치가 된다. 특히 회로 내의 신호선들을 서로 독립적인 것으로 가정하고 있다. 그러나 대부분의 회로들은 재수렴 팬아웃을 갖고 있기 때문에 신호선들은 서로 종속적일 수 밖에 없다.

본 논문에서는 신호선 상관관계를 고려한 개선된 테스트용이도 분석 알고리즘을 제안하였다. 제안된 테스트용이도 분석 알고리즘은 재수렴 팬아웃 정보를 분석하여 저장한 후에 이를 SCOAP에서 반영하여 테스트용이도를 계산하도록 처리함으로써, 기존의 SCOAP보다 정확한 테스트용이도를 계산할 수 있었다. 또한 재수렴 팬아웃에 대한 별도의 사전 처리를 함으로써, 재수렴 팬아웃으로 인해 제어가 불가능한 신호선을 미리 파악할 수 있게 됨에 따라, 고장검출이 불가능한 신호선에 대한 정확한 제어도를 할당할 수가 있었다. 이로 인하여 회로의 전체적인 테스트용이도가 보다 정확해질 수 있었다.

참고문헌

[1] K. T. Cheng and C. J. Lin, "Timing Driven Test Point Insertion for Full-Scan and Partial-Scan BIST", Proc. Int. Test Conf., pp. 506-514, 1995.

[2] H. C. Tsai, K. T. Cheng, C. J. Lin, and S. Bhawmik, "A Hybrid Algorithm for Test Point Selection for Scan-Based BIST", Proc. Design Automation Conf., pp. 478-483, 1997.

[3] Y. Savaria, M. Youssef, B. Kaminska, and M. Koudil, "Automatic Test Point Insertion for Pseudo-Random Testing", Proc. Int. Symp. Circuits and Systems, pp.

1960-1963, 1991.

[4] L. H. Goldstein, "Controllability/Observability Analysis of Digital Circuits", IEEE Trans. on Circuits and Systems, Vol. CAS-26, pp. 685-693, Sept. 1979.

[5] F. Brglez, "On Testability Analysis Of Combinational Networks", Proc. Int. Symp. Circuits and Systems, pp. 221-225, May 1984.

[6] R. G. Bennetts, C. M. Maunder, and G. D. Robinson, "CAMELOT: A Computer-Aided Measure for Logic Testability", IEE Proc., Vol. 128, pp. 177-189, Sept. 1981.

[7] V. D. Agrawal and S. C. Seth, "Probabilistic Testability", IEEE Int. Conf. on Computer Design, pp.562-565, Oct. 1985.

[8] K. P. Parker and E. J. McCluskey, "Probabilistic Treatment of General Combinational Networks", IEEE Trans. Comput., Vol. C-24, pp. 668-670, 1975.

[9] S. Chakravarty and H. B. Hunt III, "On Computing Signal Probability and Detection Probability of Stuck-at Fault", IEEE Trans. On Computer, Vol. 39, pp. 1369-1377, Nov. 1990.

[10] S. C. Seth, L. Pan, and V. D. Agrawal, "PREDICT-Probabilistic Estimation of Digital Circuit Testability", Fault-Tolerant Comp. Symp. pp. 220-225, June 1985.

[11] S. C. Chang, W. Jone, and S. S. Chang, "TAIR: Testability Analysis by Implication Reasoning", IEEE Trans. on Computer-Aided Design, Vol. 19. No. 1, Jan. 2000.