

論文 98-35C-4-2

플립플롭의 초기화 가능성을 고려한 디지털 회로에 대한 고장 검출율의 평가 기법

(Evaluation of Fault Coverage of Digital Circuits Using Initializability of Flipflops)

閔 炯 福 * , 金 信 澤 ** , 李 宰 勳 *

(Hyoung-Bok Min, Shin-Taek Kim, and Jae-Hoon Lee)

요 약

디지털 회로의 검사 신호에 대한 고장 검출율을 정확하게 계산하기 위해 결함 시뮬레이터를 사용한다. 그러나 결함 시뮬레이터의 실행 시간은 회로를 구성하는 게이트 수의 제곱에 비례하여 오늘날의 대규모 회로에서는 많은 CPU 시간을 소모한다. 따라서 정확도는 떨어지지만 비교적 짧은 시간 안에 회로의 검사성을 평가하기 위한 검사성 분석 알고리즘이 제안되었다. COP는 빠르고 정확하나 순차회로에 적용할 수 없으며 STAFAN은 순차회로에 적용이 가능하지만 정상 회로 시뮬레이션의 사용으로 실행시간이 많이 소모된다. 본 저자들은 정확한 고장 검출율을 단시간에 얻기 위해서 EXTASEC을 제안한 바 있다. EXTASEC은 극히 일부 회로에서 정확도가 떨어지는 현상이 발생하였으며, 이 논문에서는 EXTASEC의 문제가 플립플롭의 초기화와 관련이 있는 것을 밝히고, 개선된 알고리즘 ITEM을 제안한다. ITEM은 후진선의 반복적 계산과 초기화되지 않는 플립플롭에 대한 분석으로 고장 검출율을 정확하고 빠르게 얻을 수가 있었다.

Abstract

Fault simulator has been used to compute exact fault coverages of test vectors for digital circuits. But it is time consuming because execution time is proportional to square of circuit size. Recently, several algorithms for testability analysis have been published to cope with these problems. COP is very fast and accurate but cannot be used for sequential circuits, while STAFAN can be used for sequential circuits but needs vast amount of execution time due to good circuit simulation. We proposed EXTASEC which gave fast and accurate fault coverage. But it shows noticeable errors for a few sequential circuits. In this paper, it is shown that the inaccuracy is due to uninitializable flipflops, and we propose ITEM to improve the EXTASEC algorithm. ITEM is an improved evaluation method of fault coverage by analysis of backward lines and uninitializable flipflops. It is expected to perform efficiently for very large circuits where execution time is critical.

I. 서 론

* 正會員, 成均館大學校 電氣工學科

(Dept. of Elec. Eng., Sungkyunkwan Univ.)

** 正會員, 大林專門大學 電子計算科

(Dept. of Computer Science Daelim College of Tech.)

※ 본 연구는 1997년도 한국과학재단 연구비 지원에 의한 결과임(과제번호 96-0102-16-01-4).

接受日字:1997年12月15日, 수정완료일:1998年3月26日

오늘날 VLSI(Very Large Scale Integration)를 설계하고 검사하는 분야에서는, 시스템을 설계할 때 필요한 경비를 감소시키고 시스템의 성능과 신뢰도를 향상시키기 위하여 많은 연구가 진행되고 있다. VLSI를 검사하는 방법이나 장치가 경제적이지 못할 경우 검사에 소요되는 비용이 생산 원가에서 큰 부분을 차

지한다. 따라서 더욱 효과적이면서 정확한 검사를 수행하기 위해 생산에 앞서 설계와 검사 계획을 먼저 수립한다. 검사 계획에서는 검사성 분석(testability analysis)과 검사 신호(test vector)의 생성이 가장 중요하다.

VLSI에 대한 검사는 검사 신호를 회로에 입력한 후 결함으로 인한 현상을 회로의 최종 출력(primary output: PO)에서 관찰하는 것이다. 즉 임의의 검사 신호에 대하여 결함이 존재할 때와 존재하지 않을 때의 최종 출력이 서로 달라야 그 결함을 외부에서 검출할 수 있다. 이때 결함을 모형화 하는 방법으로 고착 고장이 널리 사용된다^[11]. 한편 검사하려는 회로가 조합 회로(combinational circuit)인 경우 한 개의 검사 신호로 고착 고장을 검사하나, 순차 회로(sequential circuit)의 경우는 연속적인 검사 신호가 요구된다. 따라서 조합 회로에 대한 결함 해석보다 순차 회로에 대한 해석이 더 복잡하다^[2].

회로에 고장이 발생했을 때 임의의 검사 신호로 그 고장을 검출할 수 있는 확률을 고장 검출율(fault coverage: FC)로 정의한다. 주어진 시험벡터에 대한 고장 검출율을 정확하게 계산하기 위해 결함 시뮬레이터(fault simulator)를 사용한다. 그러나 결함 시뮬레이터의 실행 시간이 회로를 구성하는 게이트 수의 제곱에 비례하므로 VLSI의 집적도가 높아질수록, 실행 시간은 급격하게 증가하고 이런 특성이 큰 제한 요소로 작용한다. 이에 대한 대책으로 발생 가능한 결함 중 일부만을 분석하기도 하지만, 이 방법으로는 선택되지 않은 결함에 대한 자료 획득이 불가능하다. 이런 문제점들을 해결하기 위해 근사값을 사용하거나 확률의 개념을 도입하는 방법이 연구되고 있으며 대표적인 기법으로 COP^[3]와 STAFAN^[4]이 있다. COP는 계산이 간단하며 매우 빠르지만 순차회로에 적용할 수 없고, STAFAN은 순차회로에 적용이 가능하나 정상 회로 시뮬레이션을 이용하기 때문에 계산시간이 많이 걸리는 단점이 있다.

이에 본 연구자들은 순차 회로를 해석할 수 있으며 회로에 대한 고장 검출율을 정확하고 신속하게 계산하는 기법 EXTASEC(Extension of Testability Analysis for SEquential Circuits)을 제시한바 있다^[5]. 제안되었던 기법은 순차 회로를 고속으로 해석하기 위해 COP와 STAFAN을 통합 확장하였다. COP의 계산식을 응용하여 회로의 각 신호선에 대한 제어

율과 관측률을 계산하고, STAFAN의 루프 민감률(loop sensitivity)을 응용하여 순차 회로에 존재하는 후진선(backward line)을 반복적으로 해석하였다. 그러나 EXTASEC은 극히 일부 회로에서 부정확한 고장 검출을 예측을 하였다.

본 논문에서는 5개의 ISCAS89 회로에서 EXTASEC으로 계산한 결과가 부정확한 이유가 회로 내의 플립플롭의 초기화와 관련이 있다는 것을 밝히고, 부정확한 고장 검출을 평가치를 주는 문제점을 해결하였으며 그 방법은 다음과 같다.

첫째, 순차 회로에서 발생하는 논리 X의 작용을 반영하기 위해 X-제어율을 도입하였다. 즉, COP의 계산식으로는 논리 X를 처리할 수 없으므로 각 신호선에 대하여 X-제어율을 추가적으로 계산한다.

둘째, 조합회로의 경우 COP와 EXTASEC은 속도 및 고장 검출율이 동일하였고, 순차회로의 경우는 고장검출도는 STAFAN과 대부분 동일하였고 속도는 월등히 고속이었다. 그러나 순차 회로에서 신호가 재수렴하는 현상 때문에 초기화되지 않는 플립플롭이 발생할 수 있다. 이 경우 X-제어율을 도입하였을 지라도 반복적인 방법으로는 이들 플립플롭들의 작용을 효과적으로 처리할 수 없기 때문에 몇몇 회로의 경우 STAFAN에서 얻은 고장 검출율과 큰 차이를 나타내었다. 이 문제의 해결을 위해서 본 논문에서 기술하는 고장 검출율 분석 알고리즘인 ITEM(Iterative Method for Testability Measurement)에서는 초기화되지 않는 플립플롭을 검색하고 검색된 플립플롭에 X-제어율을 할당함으로써 정확한 고장검출도를 얻을 수 있었다.

본 논문의 구성은 2장에서 선행연구의 EXTASEC을 설명하였고 X-제어율과 EXTASEC이 가지는 문제점을 제시하였다. 3장에서는 EXTASEC의 문제점이 초기화되지 않는 플립플롭으로 인한 것임을 밝히고 이를 반영한 ITEM의 정확도를 4장에서 ISCAS85 조합 회로와 ISCAS89 순차 회로에 대하여 검증하였으며 5장에서 결과를 나타내었다.

II. EXTASEC

본 장에서는 선행 연구결과인 EXTASEC^[5]을 요약하고 문제점을 제기한다. 또한 초기화되지 않는 플립플롭으로 인한 영향을 반영하기 위해 새로이 X-제

어울을 도입하였다. 본 논문에서는 EXTASEC을 기반으로 하여 문제점을 보완하여 만들어진 새로운 알고리즘을 ITEM이라 부르기로 한다.

ITEM은 계산속도는 빠르지만 순차회로에 적용이 곤란한 COP와 속도는 느리지만 순차회로를 해석할 수 있는 STAFAN을 서로 보완한다. 즉 순차회로에 대한 해석이 가능하면서도 계산속도를 빠르게 하기 위해 순차회로에 포함된 후진선을 반복적으로 분석한다.

1. 제어율, 관측률, 검출률과 X-제어율

회로의 최초 입력에 임의의 검사 신호를 인가했을 때 내부의 신호선 l 이 어떤 논리 값으로 제어될 확률을 그 신호선의 제어율이라 한다^[6]. 그리고 논리 1과 0에 대한 제어율을 $C1(l)$ 과 $C0(l)$ 로 각각 표기한다. 그리고 신호선 l 의 현재 값(논리 1, 0, 또는 X를 구분하지 않음)이 회로의 최종 출력에서 관측될 확률을 그 신호선의 관측률이라 한다. 그리고 이것을 $OB(l)$ 로 표기한다.

신호선 l 에 대하여 검출률 $D1(l)$ 과 $D0(l)$ 을 정의한다. 검출률의 의미는 신호선에 존재하는 stuck-at-1 이나 stuck-at-0의 결함을 검사 신호로 검출할 확률이다.

$$D1(l) = C0(l) * OB(l) \tag{1}$$

$$D0(l) = C1(l) * OB(l) \tag{2}$$

이때 각 게이트들의 논리 동작에 따라 제어율과 관측률의 계산식을 결정하며, 기본게이트에 대한 계산식은 COP와 동일하다. 그리고 COP에서 정의하지 않은 게이트인 XOR와 플립플롭의 제어율과 관측률을 사용하였다.

이에 본 논문에서는 순차 회로에서 발생하는 논리 X를 반영하기 위해 X 제어율을 도입하였다. X-제어율은 EXTASEC에서는 없던 개념이다. 이것은 검사 신호를 회로에 인가했을 때 신호선 l 이 0또는 1로 제어되지 않을 확률을 의미하며, 회로 내의 모든 신호선에 대하여 $CX(l)$ 를 식 (3)로 계산한다.

$$CX(l) = 1 - C1(l) - C0(l) \tag{3}$$

2. 후진선의 검사성 계산

게이트와 네트로 구성된 회로를 깊이 우선 탐색(depth first search) 기법으로 검색한다. 그리고 각 신호선을 가지선, 후진선, 전진선, 횡단선의 4 가지로

분류한다^[5]. 이때 순차 회로에 포함된 후진선을 분석하기 위해 반복법을 사용한다.

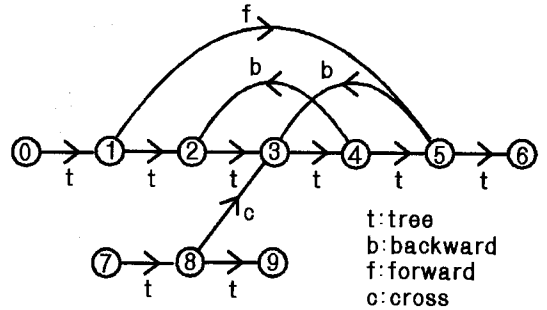


그림 1. 신호선의 분류
Fig. 1. Classification of lines.

후진선에 의해 루프가 형성되며, 그림 1에서 후진선 4-2와 5-3에 의해 루프 4-2-3-4와 5-3-4-5가 형성되었다.

루프를 구성하는 신호선들에 대하여 필요한 확률을 계산하기 위해 일반적으로 다음 두 가지 방법을 사용한다. 첫째, 루프를 구성하는 게이트를 추적하면서 각 신호선들의 확률을 계산한다. 이때 모든 신호선에서 확률 값의 변화가 없을 때까지 추적 및 계산을 계속한다. 이 방법을 적용하기 위해서는 루프를 구성하는 게이트의 입력 값들이 모두 독립적이라는 가정이 필요하다. 그러므로 한 신호가 서로 다른 경로를 통하여 동일한 루프의 다른 곳으로 연결될 경우에는 사용이 부적당하다. 둘째, 각 루프에 대한 민감률을 이용한다. 즉, 루프를 구성하는 게이트들의 입력 신호선 중에서 루프를 구성하는 신호선들을 제외한 나머지 신호선들을 모두 찾고, 그 신호선들이 해당 게이트를 민감하게 하는 확률을 먼저 계산한다. 그러나 다수의 후진선이 서로 간섭을 하면서 루프를 형성하는 경우, 회로 내의 모든 루프를 검색하는데 소요되는 실행 시간이 크게 증가한다.

먼저 그림 2의 모든 후진선에 대하여 점선으로 표시한 부분을 절단한다. 그리고 절단된 후진선 b 에 대하여 최초 제어율을 $C1_0(b)$, $C0_0(b)$, 그리고 $CX_0(b)$ 라 하면, 플립플롭의 초기 상태에 의해 확률 $CX_0(b)$ 는 1이 되고 식 (3)에 의해 확률 $C1_0(b)$ 와 $C0_0(b)$ 는 모두 0이 된다. 계산 속도 상의 이런 문제점을 해결하기 위해 본 연구에서는 다음과 같은 반복 계산법을 제시한다.

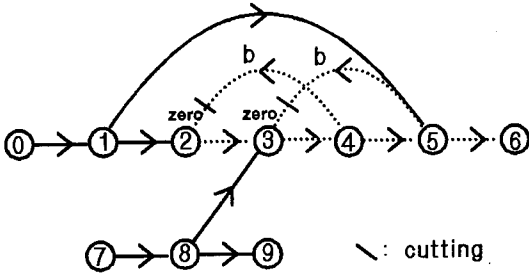


그림 2. 제어율의 오차 확산
Fig. 2. Propagation of invalid controllability.

최초 입력에서 시작하여 최종 출력이나 절단된 후진선까지 게이트를 검색하면서 각 신호선의 제어율을 계산한다. 그러면 절단된 신호선 b 에 대하여 새로운 제어율 $C_{11}(b)$, $C_{01}(b)$, 그리고 $CX_1(b)$ 가 계산된다. 이때 앞서 가정한 초기값 $C_{10}(b)$, $C_{00}(b)$, 그리고 $CX_0(b)$ 들은 검사 신호가 계속 입력되는 상황을 반영하지는 못한다. 따라서 후진선의 현재 계산된 제어율 $C_{11}(b)$, $C_{01}(b)$, $CX_1(b)$ 과 그림 2에서 점선으로 표시된 신호선들의 제어율에는 모두 오차가 포함되었다. 그러나 $C_{11}(b)$, $C_{01}(b)$, 그리고 $CX_1(b)$ 들을 이용하여 각 게이트의 제어율을 동일한 방법으로 다시 계산하면 오차가 감소한다. 이때 첨자는 반복 횟수를 나타낸다. 이런 과정을 반복하면서 제어율을 계산한다.

한편 최초 입력에서 시작하여 최종 출력 쪽으로 게이트를 차례로 검색하기 위해, topological sorting을 수행한다. 위에서 설명한 여러 과정들을 통합하여, 회로의 각 신호선에 대한 제어율을 계산하는 순서는 그림 3과 같다.

Procedure Compute_Controllability(I_LIMIT)

```

begin
  Build a circuit digraph GRAPH(VERTEX, LINE).
  Cut_Backward_lines(GRAPH)
  T_SORT = Topological_Sorting(VERTEX)
  Initiate_Controllabilities(VERTEX)
  Controllabilities(I_LIMIT, T_SORT)
  return
end

```

그림 3. 제어율의 계산
Fig. 3. Computation of controllability.

한편 각 신호선의 관측률을 계산하기 위해, 그림 2의 그래프에서 각 신호선의 방향을 반대로 하여 후진선을 검색하여 제어율에서와 같이 점선으로 표시한 부

분을 절단한 후 절단한 곳의 최초 관측률 $OB_0(b)$ 를 0으로 한다. 그리고 제어율을 계산할 때 적용한 반복 과정을 동일하게 적용한다.

I_LIMIT는 제어율과 관측률을 계산할 때 적용된 반복과정의 횟수를 의미한다. I_LIMIT는 실험적으로 결정되며 I_LIMIT를 충분히 크게 하면 모든 신호선들의 제어율과 관측률이 일정한 수치에 수렴하지만, 플립플롭은 신호선 clk 에 상승 신호가 있을 때 동작하기 때문에 I_LIMIT를 제한해야 하는데 I_LIMIT=60일 때 결합 시뮬레이터의 결과에 가장 근접한다.

3. EXTASEC의 결과 및 문제점
EXTASEC의 전체 구성을 그림 4에 나타내었다.

Procedure EXTASEC(I_LIMIT)

```

begin
  Build a circuit digraph GRAPH(VERTEX, LINE).
  Build a digraph R_GRAPH(R_VERTEX, R_LINE)
  whose edges have opposite directions to GRAPH.
  Cut_Backward_lines(GRAPH)
  Initiate_Controllabilities(VERTEX)
  T_SORT = Topological_Sorting(VERTEX)
  where T_SORT is a list of sorted vertices
  Controllabilities(I_LIMIT, T_SORT)
  Initiate_Observabilities(VERTEX)
  R_T_SORT = Topological_Sorting(R_VERTEX)
  Observabilities(I_LIMIT, R_T_SORT)
  Compute fault coverage.
  return
end

```

그림 4. EXTASEC의 구조
Fig. 4. Structure of EXTASEC.

ISCAS85 조합 회로의 일부 회로에 적용한 결과를 그림 5(a)에 나타내었다. ISCAS89 순차 회로에 대한 결과를 그림 5(b)에 나열하였다. 그림에서 회로 s5378의 경우, 계산의 정밀도 면에서 제안한 반복법이 STAFAN보다 더 정확하다.

제안한 반복법의 실행 시간을 STAFAN과 비교하여 그림 6에 나타내었다. 이것은 크기가 6000 [개]인 랜덤 입력에 대한 실행 시간의 비교이다.

결과적으로 EXTASEC에서 계산한 고장 검출율을 결합 시뮬레이터에 비교한 결과 서로 근접한다. 순차 회로의 경우에도 계산의 정밀도 면에서 기존의 다른 기법들에 필적하고 일부 회로에서는 더 정확한 결과를

나타내기도 한다. 회로의 고장 검출율을 계산하기 위해 결합 시뮬레이터나 STAFAN을 사용하면, 회로의 복잡도가 커짐에 따라 실행 시간이 느려진다. 그러나 제안한 EXTASEC을 사용하면, 순차 회로에 대한 결합 해석이 가능하고 또 실행 시간이 현저하게 단축된다.

III. 플립플롭의 초기화 가능성

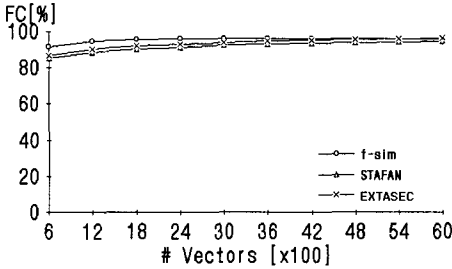
일부 순차 회로에서는 재수렴하는 신호에 의해 결합이 존재할 때 초기화되지 않는 플립플롭 UIFF (uninitializable flipflop)^{[7] [8]}가 발생하며 이들은 고장 검출율의 정확성에 영향을 미친다.

1. 플립플롭의 초기화

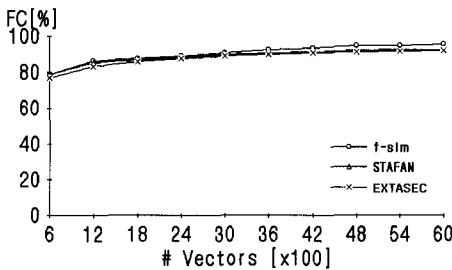
순차 회로 s208의 일부분을 그림 7(a)에 표시하였다. 그림에서 소수점 두 자리의 숫자는 상단과 하단이 제어율 C1과 C0을 각각 의미한다. 그림에서 플립플롭 X.1은 루프에 저장된 논리 X로 인하여 초기화되지 않는다. 따라서 회로 s208의 고장 검출율은 매우 낮은 수치를 나타낸다. 이때 플립플롭 X.1이 초기화되지 않는 이유는, FANOUT에서 분기된 신호가 다시 재수렴 한 후 서로 간섭하고 또 플립플롭의 입력에서 논리 X가 계속 유지되기 때문이다.

제안한 반복법을 동일한 회로에 적용한 결과를 그림 7(b)에 표시하였다. 그림 7 (a)와 (b)를 비교하면, 재수렴 신호들에 의해 발생된 오차가 플립플롭 X.1의 출력 신호선에서 쉽게 관찰된다. 그리고 그 오차가 최초 입력에 근접한 곳에서 발생하였으므로, 반복법을 적용하면 전체 회로에서 계산한 고장 검출율은 많은 오류를 포함한다. 이때 그림 7(c) 처럼 재수렴 영역 R3을 XOR로 변환하거나 플립플롭 X.1의 제어율을 C1=0, C0=0, 그리고 CX=1로 고정한 후 제안한 반복법을 적용하면, 본래 회로인 그림 7(a)의 제어율에 일치한다^[9].

그림 8은 회로에 존재하는 논리 X가 제거되는 경우와 그렇지 않은 경우를 나타내고 있다. 그림 8 (a)는 재수렴하는 경로를 갖지 않는 트리 구조의 회로를 나타내고 있다. 이 회로에서는 한 개의 입력 신호선을 통하여 논리 X가 인가될지라도, 나머지 신호선들에 입력되는 논리 0이나 1로 인해 그 논리 X가 제거될 수 있다. 즉, 논리 X가 진행되는 경로에 포함된 OR 게이트나 AND 게이트의 입력에 각각 논리 1이나 논리 0을 발생시킴으로 논리 X가 제거된다. 한편 그림 8 (b)는 재수렴하는 경로가 내부에 존재하는 회로를 나타낸다. 이런 회로에서는 한 개의 입력 신호선을 통하여 논리 X가 인가되는 경우, 나머지 다른 입력 신호선들에 논리 1이나 0이 인가될지라도 회로에서 논리 X가 존속할 수 있다. 그림 7 (a)에 표시된 순차 회로 s208의 경우가 그 예이다.



(a) 조합 회로의 고장 검출율 변화(c3540)



(b) 순차 회로의 고장 검출율 변화(s1196)

그림 5. 조합회로와 순차회로의 고장검출도 변화
Fig. 5. Fault coverage variations on combinational circuit and sequential circuit.

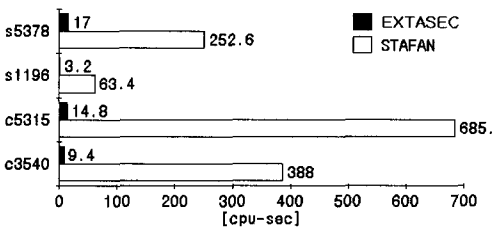
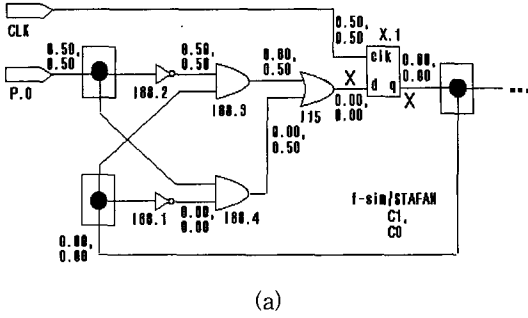


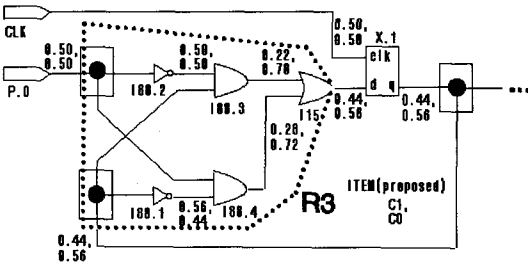
그림 6. 실행 시간의 비교

Fig. 6. Comparison of execution times.

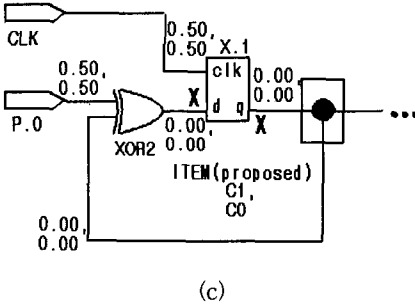
그러나 일부 순차회로(s208, s420, s838, s13207, s38417)에서는 L_LIMIT를 60으로 하였을 경우 결합 시뮬레이션이나 STAFAN과 많은 차이를 나타내었다. 분석 결과 순차 회로에서 신호의 재수렴 현상 때문에 초기화되지 않는 플립플롭이 존재할 경우 오차가 발생하였다.



(a)



(b)



(c)

그림 7. 플립플롭의 초기값

(a) 원래 회로 (b) 반복법의 오류 (c) 초기화 후 반복법

Fig. 7. Initial value of flipflop.

(a) Original circuit (b) Error of iterative method (c) Iterative method after initialization

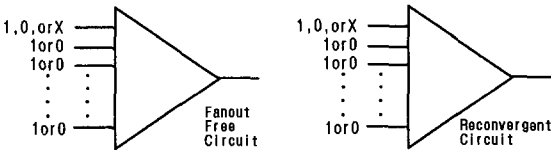


그림 8. 논리 X의 제어

(a) 제거 가능 (b) 제거 불가능

Fig. 8. Controlling logic X.

(a) Eliminable. (b) Not-eliminable.

그림 9로 표현되는 회로의 경우, 논리 X가 후진선을 통하여 인가되는 회로의 특성에 따라, 플립플롭이

초기화될 수도 있고 논리 X가 계속 유지된 상태에서 초기화되지 않을 수도 있다.

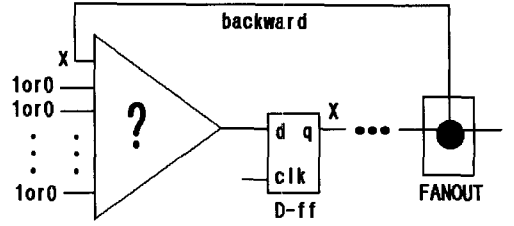


그림 9. 플립플롭의 입력

Fig. 9. Input of flipflop.

따라서 그림 7에서 반복법의 오차를 제거하기 위해 초기화되지 않는 플립플롭을 먼저 검색한다.

2. 초기화되지 않는 플립플롭

고장검출도의 정확도를 계산하기 위해 초기화되지 않는 플립플롭들을 먼저 검색한다. 검색 알고리즘은 [7]에 기술된 것을 사용하였다. 그리고 검색된 플립플롭의 출력 신호선에 대하여 제어율을 고정 한 후 제안한 반복법을 회로에 적용한다. 회로의 각 신호선에 나타나는 현재 값이 특정한 값으로 결정되지 않을 경우, 이를 효과적으로 처리하기 위해 다음의 논리 값을 사용한다^[7].

- 논리 X: 논리 0 또는 1.
- 논리 TX: 논리 X가 아님.
- 논리 TXc: 논리 TX의 보수.
- 논리 T0: 논리 0이 아님.
- 논리 T1: 논리 1이 아님.

순차 회로에서는 회로에 존재하는 논리 X가 제거되지 않을 수 있다. 그림 10에서 S에 0이 인가될 때, A에 의해 플립플롭은 논리 1이나 논리 0으로 초기화된다. 그러나 S에 1이 인가되면 플립플롭이 초기화되지 않고 그 출력단에서 논리 X가 계속 유지된다.

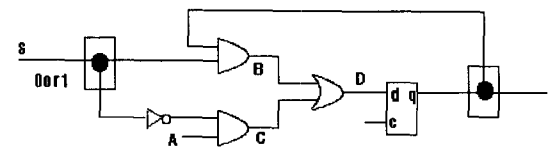


그림 10. 논리 X의 전파

Fig. 10. Propagation of logic X.

일반적으로 회로 내부의 임의의 신호선을 특정 논리 값으로 고정할 때, 플립플롭의 논리 X가 다시 입력단

으로 전파되는 경로가 존재하면 그 플립플롭은 초기화 되지 않는다.

초기화되지 않는 플립플롭을 검출하기 위해, 플립플롭의 최초 출력 값인 논리 X나 논리 TX가 후진선을 통하여 다시 입력으로 전달되는지 먼저 조사한다. 이때 플립플롭의 출력과 입력간의 경로를 PODEM과 유사한 방법으로 검색한다.

전진 함축(forward implication)에서 해당 게이트의 입력에 따라 그 출력을 계산하고, 후진 함축(backward implication)에서 출력에 따라 입력을 결정한다. 이때 다음 조건에 따라 TX-frontier를 정의한다.

- 1) 게이트의 출력이 논리 0 또는 1로 고정되지 않는다.
- 2) 게이트의 입력이 TX 또는 TXc를 갖는다.
- 3) 게이트로부터 플립플롭까지의 경로(path)가 존재한다.

그림 11의 회로에서 플립플롭에 논리 TX를 인가하고 전진 함축을 수행하면, 그림에 표시된 바와 같이 논리 값들이 결정된다. I에 있는 TXc를 플립플롭의 입력 D로 전달하기 위하여, A에 논리 1을 배정하고 전진 함축을 수행한다. 이때 J는 TXc를, C와 G는 논리 0을 각각 갖는다. 결국 TX가 플립플롭의 입력 D로 전파되고, 플립플롭은 A의 논리 1에 의해 초기화가 불가능하다.

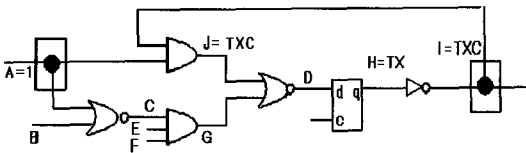


그림 11. 초기화할 수 없는 플립플롭의 예
Fig. 11. Example of Uninitializable Flipflop.

플립플롭이 A=1에 의해 초기화가 불가능하다고 판정되었으므로, 논리 1의 보수인 논리 0을 A에 배정하고 전진 함축을 수행한다. 만약 그 수행 결과에서도 플립플롭이 초기화되지 않으면 그 플립플롭을 초기화가 불가능한 것으로 최종 판정한다. 초기화가 불가능한 플립플롭을 최종적으로 검색하는 과정은 다음과 같다^[7].

3. 정확도 개선

순차 회로에서는 후진선이나 재수렴하는 선들에 의

해 또는 초기화되지 않는 플립플롭으로 인해, 회로 내부에 논리 X가 계속 잔존한다. 이런 회로를 반복적으로 해석할 경우 큰 오차가 예견된다. 이러한 오차를 제거하기 위해 초기화되지 않는 플립플롭을 먼저 검색하고 해당 플립플롭 출력의 제어율을 C1=0, C0=0, 그리고 CX=1로 고정한다. 그리고 반복법을 적용한다. 이러한 방법으로 ITEM의 정확도가 개선되었으며 전체 구성을 그림 13에 나타내었다.

Function Find_Uninitializable_Flipflop(input: flipflop)

Assign TX at the output of F and perform forward implication from F; Create a TX-frontier T;

for each logic element G in T

```

for each fanin_gate of G with logic value X {
    init_flag = Backward_Implication (fanin_gate, a
non-controlling value V of G)
    if(init_flag=NO) return(NO); /* uninitializable */
return(YES); /*initializable */

```

Function Backward_Implication(G,V)

if G is a fanout stem or a Primary input do forward implication with V;

if the input of F is either TX or TXc

return(NO) /* uninitializable */

else undone the implicated values;

if G is a Primary Input /* recursive call */

Backward_Implication(fanin_gate of G, a logic value to justify

V at G);

return(YES);

그림 12. 초기화되지 않는 플립플롭의 검색
Fig. 12. Search for uninitializable flipflop.

Procedure ITEM(I_LIMIT)

begin

Build a circuit digraph GRAPH(VERTEX, LINE).

for every v ∈ VERTEX

if v is flipflop

Find_Uninitializable_Flipflop(v)

Modify X-controllability

endif

EXTASEC(I_LIMIT)

return

end

그림 13. ITEM의 구조

Fig. 13. Structure of ITEM.

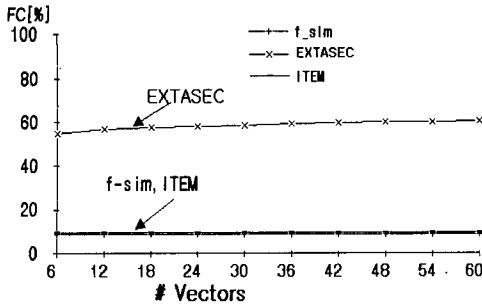


그림 14. ITEM의 정확도
Fig. 14. Accuracy in ITEM.

개선된 ITEM을 s208 회로에 적용한 결과를 그림 14에 표시하였다. 앞에서 살펴본 바와 같이, 회로 s208에서 초기화되지 않는 플립플롭이 발견되며 그 원인은 FANOUT에서 분기된 신호가 다시 재수렴하여 논리 X가 플립플롭의 입력에서 계속 유지되기 때문이다. 따라서 초기화할 수 없는 플립플롭을 고려하지 않고 반복법을 수행한 결과 큰 오차가 발생한다. 이것은 s208 회로 내부에 존재하는 초기화할 수 없는 5개의 플립플롭에 기인한 것이다. ITEM은 이런 플립플롭들에 대하여 그 출력 신호선의 제어율을 C1=0, C0=0, 그리고 CX=1로 고정한 후 반복법을 적용한다. 따라서 ITEM은 그 계산 결과가 더 정확하며, 그림 14에서 결합 시뮬레이션에 가까이 접근함을 알 수 있다.

IV. 제안된 기법 ITEM의 검증

본 장에서는 제안한 기법 ITEM의 유용성을 검증하기 위해, ISCAS85 조합 회로와 ISCAS89 순차 회로를 사용하여 고장 검출율의 정확성과 실행 시간의 단축을 비교 분석한다. 특히 순차 회로에서 초기화할 수 없는 플립플롭이 발생했을 때, 그 플립플롭을 고려한 경우와 고려하지 않은 경우를 모두 분석하였다.

조합 회로에 대하여 랜덤 입력 6000 [개] 를 적용하여 계산한 고장 검출율과 실행시간을 표 1에 나열하였다. 표에 나타난 것처럼 ITEM의 계산 결과가 본 연구실에서 개발한 결합 시뮬레이터(F-SIM)의 계산 결과에 근접한다. 조합 회로의 경우 ITEM과 STAFAN은 고장 검출율이 결합 시뮬레이터의 결과와 거의 일치하여 정확도가 비슷하지만 표에 나타난 것처럼 ITEM의 실행 시간이 STAFAN 보다 고속이

다. ITEM은 STAFAN이나 결합 시뮬레이터처럼 검사 신호를 사용하지 않기 때문에 계산을 크게 단축시킬 수 있다.

표 1. 조합 회로의 고장 검출율
Table 1. Fault coverage on combinational circuits.

CLK	고장 검출률[%]			실행시간[cpu-sec]	
	F-SIM	STAFAN	ITEM	STAFAN	ITEM
C432	98.84	99.38	100.00	35.6	1.1
C499	99.19	100.00	100.00	37.4	1.3
C880	99.60	99.38	99.63	64.5	2.2
C1355	99.70	99.94	98.85	112.3	3.6
C1908	99.65	99.45	99.22	205.5	5.0
C2670	84.01	86.03	87.20	278.1	7.2
C3540	96.20	94.49	96.27	388.0	9.4
C5315	99.41	99.95	99.66	685.4	14.8
C6288	99.45	99.58	100.00	6094.0	17.5
C7552	94.30	99.86	95.00	1061.1	21.0

순차 회로에서 초기화되지 않는 플립플롭을 고려하지 않고 반복법으로만 계산한 고장 검출율과 실행시간, 그리고 초기화되지 않는 플립플롭을 고려한 고장 검출율과 실행시간을 표 2에 나열하였다. 이때 사용한 랜덤 입력의 갯수는 6000 [개] 이다.

EXTASEC에서는 순차회로를 반복적으로 해석할 때 반복횟수(L_LIMIT)를 60으로 적용하였으나 몇몇 회로(표 2에서 * 표시 회로)에서는 결합 시뮬레이터와 많은 차이를 보였었다. 그 원인은 이 회로들내에 초기화 할 수 없는 플립플롭을 다수 포함하여 이들로 인해 오차가 발생하였기 때문이다. 본 연구에서는 오차를 제거하기위해 초기화되지 않는 플립플롭을 검색하고 검색된 플립플롭의 출력 신호선에 대하여 제어율을 고정한다. 그리고 ITEM을 회로에 동일하게 적용한다. ITEM에서는 초기화할 수 없는 플립플롭들의 제어율을 C1=0, C0=0, 그리고 CX=1로 고정한 후 계산을 수행한다. 표 2에서 초기화되지 않는 플립플롭들이 존재하는 회로(UIFFs가 0이 아닌 회로)는 초기화를 고려한 것과 고려하지 않는 것 사이에 큰 오차가 발생되며 초기화가 불가능한 플립플롭이 고려된 ITEM의 결과(Initialized)가 결합 시뮬레이터에 근접하는 결과를 볼 수 있다. 그러나 초기화되지 않는 플립플롭이 없는 회로(UIFFs가 0인 회로)의 경우는 결과가 동일하다.

특히 크기가 큰 회로(회로 S38417 또는 S35932 등)에서는 더 정확한 분석을 위해 검사 신호의 크기를 증가시켜야 하며, 이 경우 ITEM을 결합 시뮬레이터에 더욱 근접시킬 수 있다.

표 2. 순차 회로의 고장 검출율
Table 2. Fault coverage on sequential circuits.

UIFFs : 초기화되지 않은 플립플롭의 수

CKT	고장 검출률(%)				실행 시간(cpu-sec)			UIFFs
	F-SIM	STAFAN	EXTASEC	ITEM	STAFAN	EXTASEC	ITEM	
S27	100.00	93.33	100.00	100.00	1.9	0.1	0.1	0
*S208	8.71	8.36	60.65	9.40	10.6	0.6	0.6	5
S298	80.36	78.34	85.97	85.97	10.7	0.9	1.5	0
S344	97.31	94.28	96.66	96.66	14.3	1.1	1.4	0
S386	71.63	68.35	75.27	75.27	15.6	1.3	3.8	0
*S420	6.55	6.34	31.63	7.53	21.7	1.0	1.4	11
S510	0.00	0.00	0.00	0.00	17.4	0.7	8.8	0
S526	9.88	8.66	10.50	10.50	15.5	0.8	3.3	0
S713	83.38	83.46	88.15	88.15	42.3	2.4	2.4	0
*S838	5.91	5.77	18.17	5.11	49.8	1.9	4.5	23
S953	7.97	8.06	8.29	8.29	44.1	1.4	2.9	0
S1238	92.20	90.74	87.02	87.02	64.8	3.3	3.4	0
*S13207	12.06	11.98	15.32	13.57	612.3	33.0	532.0	8
S35932	86.14	89.63	91.14	91.14	2549.0	145.9	948.7	0
*S38417	4.11	4.09	24.77	9.99	1676.7	127.2	1315.8	2

V. 결론

본 논문에서는 디지털 회로에 대한 고장 검출율을 계산하는 EXTASEC 기법을 개선하여 새로운 기법 ITEM을 제안하였다. ITEM은 순차 회로에 대하여 검사성을 정확하고 빠르게 분석하며 다음과 같은 특징을 갖는다.

첫째, 순차 회로에 존재하는 후진선을 반복적으로 분석하였다. COP는 계산 속도가 빠르지만 순차 회로를 분석할 수 없고, STAFAN은 속도가 느리지만 순차 회로를 해석할 수 있다. 순차 회로에 대한 해석이 가능하면서 계산 속도를 빠르게 하기 위해, ITEM에서는 이들 두 기법을 서로 보완하였다. 즉, 전자의 계산식을 이용하여 각 신호선의 제어율과 관측률을 계산하며, 후자를 응용하여 순차 회로의 후진선을 반복적으로 해석하였다.

둘째, 논리 X에 대한 제어율을 도입하였다. COP에서 정의한 계산식을 이용하면 회로의 각 신호선에 대

한 제어율과 관측률이 계산된다. 그러나 이들 계산식으로는 순차 회로에서 발생하는 논리 X를 효과적으로 처리할 수 없다. 이에 순차 회로에 존재하는 논리 X의 작용을 효과적으로 분석하기 위해 각 신호선에 대하여 추가적으로 X-제어율을 계산하였다.

세째, 순차 회로에서 초기화가 불가능한 플립플롭의 제어율을 C1=0, C0=0, 그리고 CX=1로 고정하였다. 초기화되지 않는 플립플롭이 포함된 순차 회로에 위에서 제시한 반복법을 적용할 경우, 재수렴하는 신호의 작용으로 인해 오차가 발생한다. 이때 반복 횟수를 경험적으로 조정하여 계산 결과의 정확도를 높일 수 있지만, 해당 회로에 적합한 반복 횟수를 효과적으로 예측할 수 없다. 이때 초기화되지 않는 플립플롭을 검색하고 그 제어율을 고정한 후 반복법을 적용하면, 재수렴하는 신호로 인해 발생하는 오차가 제거된다.

ITEM 기법의 유용성을 검증하기 위해, STAFAN, COP, 결합 시뮬레이터, 그리고 제안한 ITEM을 ISCAS85 조합 회로와 ISCAS89 순차 회로에 적용하였다. 검증 결과 조합 회로에서 뿐만 아니라 순차 회로에 대한 결과에서도 정확도가 STAFAN과 유사하였으며, 또한 실행 시간은 STAFAN이나 결합 시뮬레이터에 비하여 단축됨을 확인하였다.

본 논문에서 제시한 개선된 ITEM 기법으로 디지털 회로에 대한 고장 검출율을 계산하면 i) 순차 회로에 대한 해석을 할 수 있으며, ii) 정확한 값을 계산할 수 있고, iii) 실행 시간을 단축시킬 수 있다.

참고 문헌

[1] V. D. Agrawal and S. C. Seth, *Tutorial: Test Generation for VLSI Chips*, Computer Society Press, Washington, D.C., 1988.

[2] A. Ghosh, S. Devadas and A. R. Newton, *Sequential Logic Testing and Verification*, Kluwer Academic Publishers, pp. 11-55, Netherlands, 1992.

[3] F. Brglez, "On Testability Analysis of Combinational Networks," *Proceedings of the International Symposium on Circuits and Systems*, pp. 221-225, May 1984.

[4] S. K. Jain and V. D. Agrawal, "Statistical Fault Analysis," *IEEE Design & Test of*

Computers, Volume 2, pp. 38-44, February 1985.

[5] 김신태, 민형복, "순차 회로를 위한 검사성 분석법의 확장," 전자공학회 논문지, 제32권 A편 제 4호, pp. 75-84, 1995년 4월

[6] J. E. Stephenson and J. Grason, "A Testability Measure for Register Transfer Level Digital Circuit," *Digest of Papers of International Symposium on Fault-Tolerant Computing*, pp. 101-107, June 1976.

[7] 박은세, 민형복, "합수 및 위상 분석을 이용한 부

분 스캔 플립플롭 선택 알고리즘", 대한전자공학 회 CAD 및 VLSI 설계연구회 학술발표회 논문집, 1996년 5월, pp. 37-42

[8] M. Abramovici and P. Parikh, "Warning: 100% Fault Coverage May be Misleading!!," *Proceedings of International Test Conference*, pp. 662-668, 1992.

[9] N. B. Hamida and B. Kaminska, "Hierarchical Functional Testability Measure," *Proceedings of 2nd European Test Conference*, pp. 327-332, April 1991.

저 자 소 개

閔 炯 福(正會員) 第 32卷 A編 第 4號 參照
 현재 성균관대학교 전기 전자 및 컴
 퓨터공학부 부교수

金 信 澤(正會員) 第 32卷 A編 第 4號 參照
 현재 대림전문대학 전자계산과 조교
 수



李 宰 勳(正會員)
 1964년 6월 11일생. 1991년 2월 성
 균관대학교 공과대학 전기공학과 졸
 업(공학사). 1993년 2월 성균관대학
 교 공과대학 전기공학과 졸업(공학석
 사). 1993년 1월 ~ 1996년 2월 LG
 전자기술원 주임연구원. 1996년 3월
 ~ 현재 성균관대학교 전기공학과 박사과정 재학. 주관
 심분야는 VLSI CAD/Testing