

論文98-35C-3-1

대규모 집적회로 설계를 위한 무고정 부분 스캔 테스트 방법 (No-Holding Partial Scan Test Method for Large VLSI Designs)

盧 炫 澈 * , 李 東 浩 *

(Hyun-Chul Noh and Dong-Ho Lee)

요 약

본 논문에서는 대규모 집적회로 설계를 위한 부분 스캔 테스트 방법에 관하여 논한다. 제안된 방법에서는 테스트 벡터 스캔 임출력, 인가시 스캔 기억 소자나 비스캔 기억 소자의 데이터를 고정할 필요가 전혀 없다. 이 테스트 방법은 테스트 목적 설계 변형 요구와 테스트 파형이 완전 스캔 기법과 거의 동일하여 대규모 집적회로에 적용 가능하다. 널리 알려진 FAN 알고리즘을 확장하여 제안된 테스트 방법에 적합한 순차 회로 ATPG 알고리즘을 고안하였다. 아울러 제안된 테스트 방법에 효과적인 부분 스캔 알고리즘을 연구하였다. 이 부분 스캔 알고리즘은 비스캔 되었을 경우 완전 스캔 결함 검출율을 유지하는 최대 기억 요소 집합을 구한다. 실험 결과 20% 이상의 기억 소자를 스캔하지 않고도 완전 스캔의 경우와 거의 동일한 결함 검출율을 얻는 테스트를 얻을 수 있었다.

Abstract

In this paper, we propose a partial scan test method which can be applied to large VLSI designs. In this method, it is not necessary to hold neither scanned nor unscanned flip-flops during scan in, test application, or scan out. This test method requires almost identical design for testability modification and test wave form when compared to the full scan test method, and the method is applicable to large VLSI chips. The well known FAN algorithm has been modified to devise a sequential ATPG algorithm which is effective for the proposed test method. In addition, a partial scan algorithm which is effective for the proposed test method has been developed. The partial scan algorithm determines a maximal set of flip-flops which gives high fault coverage when they are unselected. The experimental results show that the proposed method allow as large as 20% flip-flops to remain unscanned without much decrease in the full scan fault coverage.

I. 서 론

최근 반도체 공정 기술의 발전과 CAD 기술의 발전으로 매우 크고 복잡한 VLSI 칩의 설계가 가능해졌다. 크고 복잡한 칩을 테스트하는 데에는 여러 가지의 테스트를 위한 설계 변형 기법 (DFT)이 요구된다. 스캔 기법은 널리 알려진 설계 변형 기법으로 모든 기억

소자가 어떤 동작 모드에서 쉬프트 레지스터로 엮여진 다. 이러한 쉬프트 레지스터를 스캔 체인이라 한다. 스캔 체인에 포함된 기억 소자를 스캔되었다고 말한다. 또한 이 쉬프트 동작 모드를 스캔 모드라 한다. 본 논문에서는 플립-플롭을 기억 소자로 사용한다 가정한다. 다른 기억 소자를 사용해도 별다른 문제는 없다. 스캔 모드를 이용하면 모든 플립-플롭을 쓰거나 읽을 수 있으므로 조합회로 ATPG를 이용하여 테스트 패턴을 생성할 수 있다. 스캔 기법을 적용한 칩은 스캔 모드와 정상 동작 모드를 번갈아 이용하여 조합회로

* 正會員, 慶北大學校 電子工學科

(Dept. of Elec. Engin., Kyungpook)

接受日: 1997年10月2日, 수정완료일: 1998年2月3日

테스트 패턴을 인가하여 칩을 테스트 할 수 있다. 조합회로 ATPG는 근래의 철저한 연구 결과 수백만 게이트 규모의 칩을 완벽하게 테스트하는 테스트 패턴을 생성할 수 있게 되었다^[1-6]. 또한 최근의 테스트 패턴 축약 기법에 대한 활발한 연구는 이들 대형 칩을 수 천개 이내의 조합 회로 테스트 패턴을 이용하여 테스트 할 수 있게 되었다^[7-10].

어떤 경우에는 설계에 주는 부담으로 인하여 완전 스캔 기법을 적용하지 못하는 경우가 있다. 이러한 경우 테스트 부담을 줄이기 위하여 부분 스캔 기법도 도입되었다. 부분 스캔 기법에서는 오직 일부분의 플립-플롭들만이 스캔된다. 부분 스캔 기법은 완전 스캔 기법과는 달리 일부분의 플립-플롭들이 스캔되지 않으므로 순차회로 ATPG를 사용하여야 한다^[11-14]. 부분 스캔 기법은 부분 스캔 선택 방법과 순차 회로 ATPG 기법으로 구성된다. 많은 저자들이 부분 스캔 선택 방법에 관하여 연구하였다^[15-28]. 대부분의 경우 부분 스캔 선택 방법을 고려하지 않은 순차 회로 테스트 생성기를 사용하였다. 이들 부분 스캔 연구는 주로 스캔할 플립-플롭을 고르는 데에 집중되었다. 많은 부분 스캔 알고리즘이 연구 문헌에 발표되었다. 이들은 주로 테스트 난이도 (testability)^[15,23], 그래프 이론 (graph theory)^[19,20,21,22], 자동 테스트 생성기 (ATPG)와의 상호 관계^[22,24,28] 등에 기반을 두고 있다.

부분 스캔 연구에 대한 활발한 연구에도 불구하고 부분 스캔 기법은 산업체에서는 널리 사용되고 있지 못하다. 이에 대한 가장 중요한 이유 중 하나는 일반적으로 순차회로 ATPG는 매우 크기가 큰 테스트를 생성하기 때문이다. 스캔 테스트의 테스트 시간과 메모리 요구가 생성된 테스트 벡터의 크기와 스캔 체인의 곱으로 나타내어짐을 고려할 때 테스트의 크기는 부분 스캔의 경우 매우 심각한 문제가 된다. 최근 순차회로 ATPG의 테스트 패턴 압축 기법이 연구되고 있다^[28,29,30]. 그러나 이 기법들은 조합회로 ATPG의 패턴 압축 기법에 비하여 성능이 열등하다^[7,8,9,10]. 학술적인 연구에서는 충분히 고려되지 못하였지만 부분 스캔 기법의 또 하나의 매우 중요한 단점은 스캔 쉬프트시 스캔되지 않은 플립-플롭의 논리값을 그전의 값으로 고정할 수 있는 기능이 추가되어야 한다는 점이다. 이를 위하여서 설계의 클록 관련 회로에 변화를 주어야 한다. 이는 면적 부담뿐만 아니라 전체 설계 시간을 증가시켜서 부분 스캔의 장점을 감소시킨다.

본 논문에서는 상기 두 문제점을 제거하여 부분 스캔 기법이 실제적으로 큰 집적회로 설계에 적용될 수 있게 한다. 이 테스트 방법은 특별히 고안된 순차 회로 ATPG 기법과 완전 스캔 결합 검출율을 크게 감소시키지 않으면서 최대한 많은 플립-플롭을 선택하게 하는 부분 스캔 기법으로 구성된다. 본 연구의 주된 기여는 새로운 부분 스캔 테스트 방법을 제안하고 간단한 실험으로 이 테스트 방법이 다수의 플립-플롭을 스캔하지 않고도 완전 스캔과 유사한 결합 검출율을 얻을 수 있음을 보이는 데에 있다. 따라서 사용된 ATPG 알고리즘에 대한 치밀한 연구와 최적의 부분 스캔 기법의 개발 등은 차후 연구 과제이다. 특히 테스트 패턴 압축에 관한 연구는 매우 흥미 있는 연구 과제이나 추후 연구 과제로 미룬다. 제안된 테스트 방법을 위한 ATPG는 전통적인 부분 스캔 테스트 방법을 위한 ATPG보다 더 어렵다. 본 논문에서는 많은 비율의 플립-플롭이 스캔되어 각각의 ATPG에 의하여 생성되는 테스트 시퀀스가 짧다고 가정한다. 대부분의 경우 70% 이상의 플립이 스캔된다. 그러나 테스트 적용 방법이 완전 스캔의 경우와 거의 같고 ATPG 문제도 완전 스캔과 유사하다. 따라서 지속적인 연구로 효과적인 ATPG 알고리즘과 부분 스캔 선택 알고리즘이 연구되면 제안된 테스트 방법은 큰 VLSI 칩의 테스트에 적용되어 완전 스캔의 면적 및 성능 부담을 획기적으로 줄일 수 있을 것으로 기대된다.

본 논문의 구성은 다음과 같다. 제 2 절은 본 논문의 주된 기여로 세 가지 부분 스캔 기법에 대하여 개념적으로 기술하고 간단한 예제 회로를 이용하여 제안된 무고정 부분 스캔 기법의 결합 검출 능력에 대하여 논한다. 제 3 절에서는 제안된 테스트 기법에 필요한 순차회로 ATPG 구현에 대하여 논한 후 완전 스캔 결합 검출율을 최대한 유지하면서 많은 플립-플롭을 스캔하지 않는 부분 스캔 선택 알고리즘을 기술한다. 제 4 절에서는 ISCAS 벤치마크 회로를 이용하여 제안된 테스트 방법의 효율을 실험한 결과를 기술한다. 제 5 절에서는 결론과 추후 연구 과제를 기술한다.

II. 부분 스캔 테스트 방법

스캔 테스트에서는 스캔 쉬프트와 테스트 벡터 적용이 주의 깊게 배열되어야 테스트 벡터의 적용과 테스

트 결과 획득이 정확하게 된다. 완전 스캔의 경우 모든 플립-플롭을 슈프트 레지스터로 엮고 스캔 모드 신호를 가하여 회로의 동작 모드에 스캔 모드와 정상 모드 두 가지를 가지게 하면 된다. 부분 스캔 기법의 경우 완전 스캔 기법과는 달리 다양한 구현 방법이 있다.

술적인 연구가 이러한 가정 하에 이루어져 왔다^[15,20,21,22]. 본 논문에서는 이 부분 스캔 방법을 일반적인 방법이라 한다.

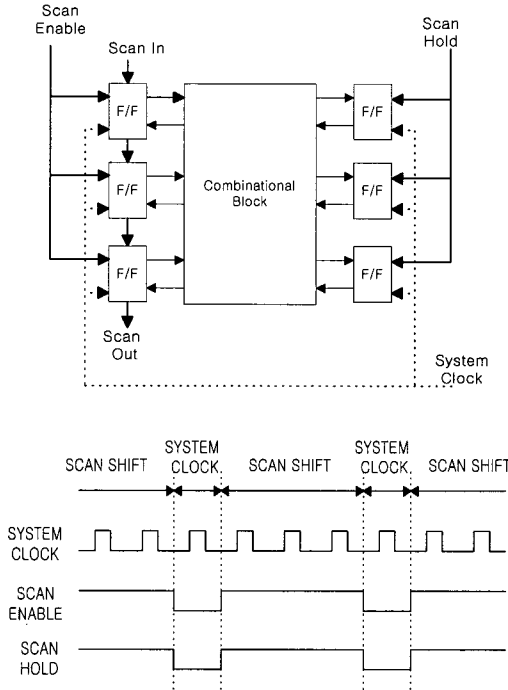


그림 1. 일반적인 부분 스캔 기법
Fig. 1. General partial scan method.

가장 일반적인 부분 스캔 기법이 그림 1에 주어져 있다. 그림 1에서 보인 바와 같이 스캔 모드 신호, Scan_Enable의 첨가뿐만 아니라 스캔되지 않는 플립-플롭에 스캔 모드에서 데이터를 고정하는 Scan_Hold 신호가 추가되어야 한다. 어떤 테스트 시퀀스의 테스트 벡터를 인가하기 위해서는 그림 1에서 보인 바와 같이 Scan_Hold 값이 1 인 상태에서 입력 벡터를 입력 스캔한다. 그 후 Scan_Hold의 값을 0으로 하여 테스트 벡터가 인가되어 회로의 기능을 작동시킨다. 작동 결과 플립-플롭에 획득된 테스트 결과가 출력 스캔된다. 출력 스캔은 다음 테스트 벡터의 입력 스캔과 병행하여 수행된다. 스캔 과정에서 스캔되지 않은 플립-플롭의 논리값을 고정할 경우 의사 입출력은 주 입출력과 테스트 생성시 동일하게 취급할 수 있다. 따라서 높은 결합 검출율을 얻을 수 있다. 거의 모든 학

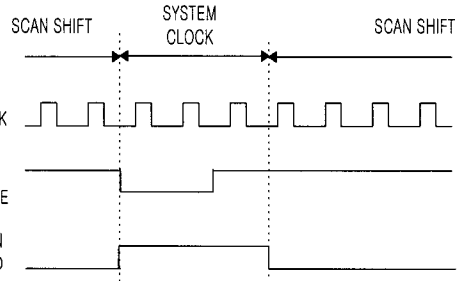
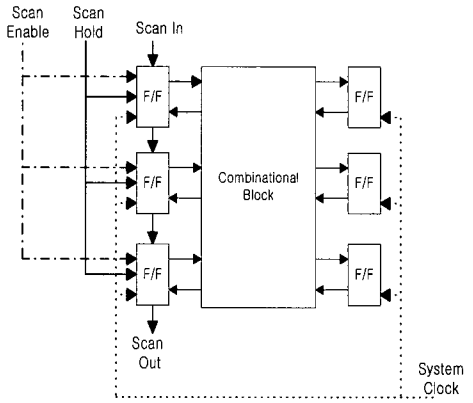


그림 2. BALLAST 부분 스캔 기법
Fig. 2. BALLAST partial scan method.

그림 2에는 Gupta 등이 제안한 BALLAST 부분 스캔 기법이 주어져 있다^[16]. 이 기법은 스캔된 플립-플롭에 데이터 고정 기능을 추가한다. 또한 많은 플립-플롭을 스캔하여 회로를 균형되게 (balanced) 만든 후 남은 플립-플롭을 버퍼로 대체하고 조합회로 ATPG를 사용하여 테스트 패턴을 생성한다. 이 방법에서는 테스트 벡터를 인가한 후 스캔된 플립-플롭의 논리값을 Scan_Hold 신호를 이용하여 고정하고 테스트 경로 상의 최대플립-플롭 수만큼 클럭을 공급하여 회로의 기능을 작동시킨다. 마지막 클럭에 스캔 플립-플롭의 고정 기능을 풀어 테스트 결과를 플립-플롭에 획득한다. 획득된 테스트 결과가 출력 스캔된다. 출력 스캔과 동시에 다음 테스트 벡터가 입력된다.

그림 3 에는 본 논문에서 제안하는 무고정 부분 스캔 기법이 주어져 있다. 그림 3 에서 보인 바와 같이 스캔 플립-플롭이나 비스캔 플립-플롭에 데이터 고정 기능을 부과하지 않는다. 이 테스트 방법에서는 테스

트 입력 벡터가 입력된 후 다수의 테스트 클록을 공급하여 결함을 활성화시키고 결함 효과를 전파시킨다. 그 후 플립-플롭들에 축적된 테스트 응답 결과를 출력 스캔한다. 스캔된 플립-플롭이 테스트 벡터인가 도중에도 고정되어 있지 않으므로 논리값이 변한다. 본 논문에서는 이 테스트 방법을 무고정 부분 스캔 방법이라 한다. 제안된 방법의 ATPG은 일반적 부분 스캔의 경우와 비교할 때에 하나의 테스트 시퀀스에서 스캔 플립에 오직 하나의 테스트 입력을 가할 수 있다는 점에서 매우 어려운 문제이다. 또한 BALLAST 부분 스캔 방법의 ATPG와 비교하면 스캔 플립-플롭의 논리치가 클록을 공급하는 도중 변화한다는 사실을 고려하여 테스트 벡터를 생성하여야 한다는 점에서 다르다. 무고정 부분 스캔 기법의 장점은 테스트를 위한 설계 변경 요구가 완전 스캔의 경우와 동일하고 테스트 파형 또한 완전 스캔과 동일하다. 다만 완전 스캔의 경우와 다른 점은 스캔 데이터가 입력된 후 다수의 시스템 클록을 공급하여 칩의 기능을 테스트한다는 점이다.

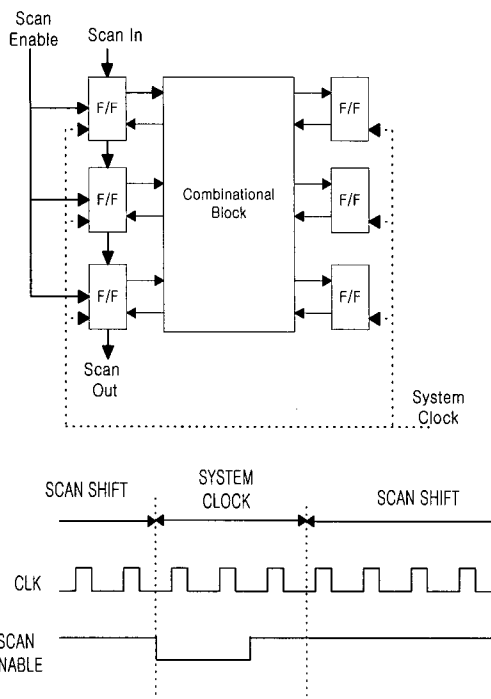


그림 3. 무고정 부분 스캔 기법
Fig. 3. No-holding partial scan method.

1. 테스트 패턴 생성 문제

먼저 위에서 언급한 세 가지 부분 스캔 테스트 방법에서 사용되는 순차 회로 ATPG 문제를 전통적인

일반적인 부분 스캔 테스트 방법, BALLAST 부분 스캔 테스트 방법, 그리고 제안된 부분 스캔 테스트 방법의 순서로 논한다.

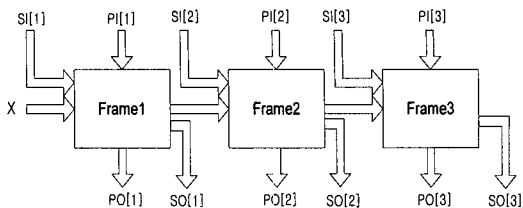


그림 4. 일반적인 부분 스캔을 위한 반복 배열 표현
Fig. 4. An iterative array representation of the circuit for the general partial scan design.

그림 4 에는 일반적인 부분 스캔 기법을 위한 반복 배열 표현이 주어져 있다. 그림 4 에서 SI [i] 와 SO [i] 는 프레임 i 의 입력 벡터와 출력 벡터를 나타낸다. S [i] 와 SO [i] 가 각 프레임에 따라 다르므로 제어도(controllability)와 관측도 (observability) 는 스캔되는 플립-플롭의 수에 비례하여 향상된다. 그러나 전술한 바와 같이 동일 테스트 시퀀스에서 다음 입력 벡터를 스캔할 때에 비스캔 플립-플롭의 논리값을 고정하여야 한다.

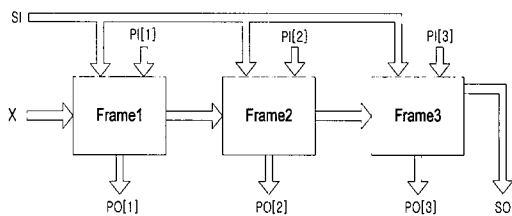


그림 5. BALLAST 부분스캔을 위한 반복 배열 표현
Fig. 5. An iterative array representation of the circuit for the BALLAST partial scan design.

그림 5에는 BALLAST 부분 스캔 기법을 위한 반복 배열 표현이 주어져 있다. 그림 4의 일반적인 방법의 경우와 비교하면 모든 프레임에 동일한 스캔 입력 벡터가 공급되고 테스트 결과는 마지막 프레임에서만 얻을 수 있다. 스캔 입력 벡터는 스캔 플립-플롭의 데이터 고정 기능에 의하여 유지된다. 이 테스트 방법의 결함 검출 능력은 일반적인 부분 스캔 테스트 방법에 비하여 떨어짐을 쉽게 알 수 있다.

그림 6 은 제안한 무고정 부분 스캔 기법을 위한 반복 배열 표현이다. 이 기법은 오직 하나의 입력 벡터

인 SI만을 스캔 입력하게 된다. 입력 벡터가 준비되면 결합 활성화와 결합 전파를 위하여 복수의 클록 펄스를 공급한다. 사용된 프레임 수가 많으면 스캔된 회로의 순차 회로 특성이 많이 남으므로 스캔 플립-플롭의 증가에 의한 테스트 용이도의 증가는 제한적이다. 그러나 많은 수의 플립-플롭이 스캔되면 테스트 용이도가 좋아져서 작은 수의 프레임으로 대부분의 결함을 검출할 수 있게 된다. 제안된 기법에서 어떤 결함에 대한 테스트 시퀀스를 생성하기 위해서는 그림 6에서 SI와 P [i] 들을 결정하면 된다. 이 테스트 생성 문제는 순차회로 테스트 생성 문제에서 단지 일부분의 플립-플롭들만 제어 불가능한 경우이다. 제안된 부분 스캔 기법의 결함 검출 능력이 일반적인 부분 스캔 기법에 비하여 떨어짐은 쉽게 알 수 있다. 그러나, BALLAST 기법과 제안된 부분 스캔 기법의 결함 검출 능력은 간단히 비교할 수 없다.

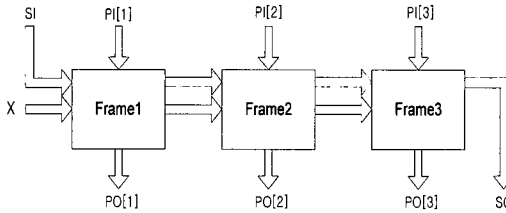


그림 6. 무고정 부분 스캔을 위한 반복 배열 표현
Fig. 6. An iterative array representation of the circuit for the no-holding partial scan design.

정리 1. 모든 동기화 순차회로에 대하여 어떤 결함이 BALLAST나 제안된 부분 스캔 테스트 방법으로 테스트되면 일반적인 테스트 방법으로 테스트 가능하다.

2. 결함 검출 능력 비교

그림 7 에는 하나의 입력을 가지고 2 개의 플립-플롭을 가진 간단한 회로가 주어져 있다. 이 회로는 플립-플롭 A가 AND 게이트로 변환되어 있으므로 1 제어도가 낮다. 완전 스캔의 경우 A, B를 모두 스캔하여 모든 고착 결함 (stuck at fault)이 검출 가능하다. 여기서는 플립-플롭 A를 스캔하는 경우를 고려한다.

그림 8은 그림 7의 회로의 반복 배열 표현 방식이 무고정 부분 스캔을 가정하여 그려져 있다. BALLAST 스캔 방식의 경우 그림 8에서 신호선 e 가

두 번째 프레임의 플립-플롭 A에 연결되지 않으며 그 대신 신호선 a 가 두 번째 프레임의 플립-플롭 A 에 입력된다. 일반적인 스캔의 경우 두 번째 플립-플롭은 의사 입력과 의사 출력이 되어 신호선 e 의 논리치를 출력하고 신호선 a_1 에 새로운 논리치를 가할 수 있게 한다. 일반적인 스캔의 경우 모든 고착 결함을 검출할 수 있음을 쉽게 알 수 있다. 본 논문에서는 대부분의 부분 스캔 연구에서처럼 고착 결함만 고려한다.

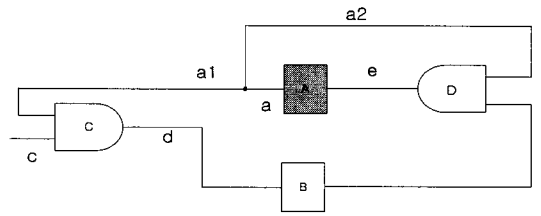


그림 7. 예제 회로
Fig. 7. Example circuit.

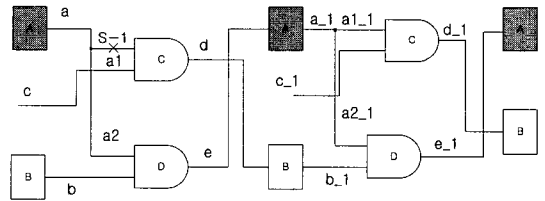


그림 8. 그림 7 회로의 반복 배열 표현
Fig. 8. An iterative array representation of Fig. 7.

나머지 두 가지 스캔 방식의 결함 검출력을 조사하기 위하여 그림 8에서 a1-s-1 결함을 고려하자. 이 결함을 검출하기 위해서는 신호 a 에 논리치 0 을 가하여야 한다. 신호 a 에 논리치 0 이 가해지면 BALLAST 스캔 방식과 무고정 스캔 방식 모두 신호 a_1 에 논리치 0 이 가해져 게이트 D를 막아 결함이 검출되지 않게 한다. 다음으로 c-s-1 결함을 고려하자. 이 결함을 검출하기 위해서는 신호 a 에 논리치 1 을 가하여 결함 효과를 전파한다. 결함을 신호 b_1부터 신호 e_1로 전파하기 위해서는 신호 a_1 에 논리치 1 을 가하여야 한다. BALLAST 스캔 방식의 경우 스캔된 플립-플롭 A 의 데이터 고정 기능에 의하여 신호 a_1 은 논리치 1 을 가지게 된다. 그러나 무고정 부분 스캔의 경우 플립-플롭 B 의 초기치가 알려지지 않으므로 신호 a_1 은 X 값을 가지게 된다. 따라서 c-s-1 결함은 BALLAST 스캔의 경우

검출이 가능하나 무고정 스캔의 경우 검출이 불가능하다.

그림 7 회로의 경우 BALLAST 스캔 방식이 무고정 스캔 방식 보다 우수한 결함 검출 능력을 가진다. 그러나 이 관계는 일반적으로 성립되지는 않는다. 다음 조합회로를 통한 극성 반전이 일어나는 회로가 그림 9 에 주어져 있다. 그림 10 는 그림 9 회로의 반복 배열 표현을 나타낸다.

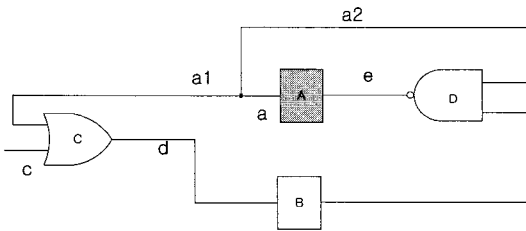


그림 9. 극성 반전 회로
Fig. 9. Polarity inversion circuit.

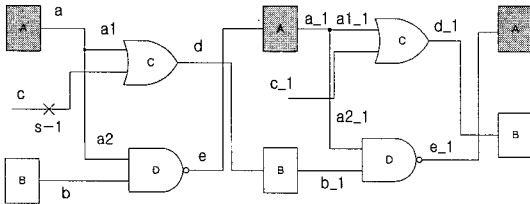


그림 10. 그림 9 회로의 반복 배열 표현
Fig. 10. An iterative array representation of Fig. 9.

그림 10 에서 $c-s-1$ 결함을 고려하자. 이 결함을 검출하기 위해서는 신호 a 에 논리치 0 을 가하여야 한다. BALLAST 부분 스캔 테스트 방법의 경우 데이터 고정 기능에 의하여 플립-플롭 A 가 논리치 0 으로 고정되어 있으므로 신호 a_1 은 논리치 0 을 가지게 되어 결함 효과가 게이트 D를 통과하지 못한다. 그러나 무고정 부분 스캔의 경우 신호 e 가 논리치 1 을 가지게 되어 신호 a_1 이 다음 프레임에서 논리치 1 을 가지게 하여 게이트 D를 통하여 결함 효과가 전파되게 한다. 따라서 무고정 부분 스캔의 결함 검출 능력은 BALLAST 부분 스캔의 결함 검출 능력과 간단히 비교되지 않는다. 다만 BALLAST 부분 스캔 방법을 위한 ATPG 문제가 무고정 부분 스캔의 경우 보다 쉽다는 것을 알 수 있다. 무고정 부분 스캔 기법을 이용하여 그림 7 의 예제 회로를 테스트하면 얻을 수 있는 결함 검출율이 매우 낮다. 다음에는 그 이유

와 높은 결함 검출율을 얻을 수 있는 테스트 목적 회로 변경 기법에 관하여 논한다. 등가 결함 중첩 알고리즘을 이용하면 10 개의 중첩 결함을 가진 결함 리스트를 얻을 수 있다. 이들 중 $a-s-0$, $a-s-1$, $e-s-1$, 그리고 $b-s-1$ 등 4 개의 결함은 결정적으로 검출된다. 나머지 6 개의 결함은 검출되지 않는다. 그 이유를 보면 $c-s-1$, $a1-s-1$, $d-s-1$, $d-s-0$ 등 4 개의 결함은 결함 효과가 플립-플롭 B 의 조절 불능으로 인하여 두 번째 프레임의 게이트 D를 통과하지 못하여 일어난다. 결함 $e-s-0$ 는 플립-플롭 B 의 조절 불능으로 신호 e 와 그 두 번째 프레임 인스턴스인 신호 e_1 을 논리치 1로 조절하지 못하여 검출이 불가능하다. 결함 $a2-s-1$ 은 결함 활성화 조건과 게이트 D를 통과하는 결함 전파 조건이 충돌되어 일어난다. 이 충돌 또한 플립-플롭 B를 논리치 1로 조절할 수 있으면 피할 수 있다. 따라서 플립-플롭 B를 논리치 1로 초기화 할 수 있으면 무고정 부분 스캔 기법을 이용하여도 플립-플롭 A 만을 스캔하고 모든 결함을 검출할 수 있다. 따라서 플립-플롭 B 가 논리치 1로 초기화 가능한 플립-플롭을 사용하거나 그림 11에서와 같은 회로 변경을 하여 완전 테스트 가능하게 할 수 있다.

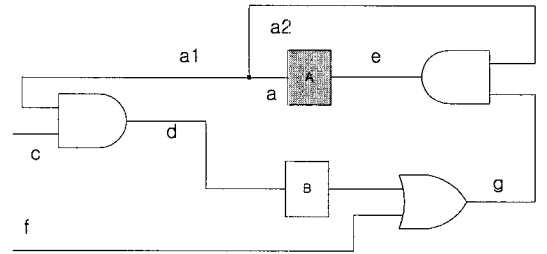


그림 11. 완전 테스트 가능한 회로
Fig. 11. Fully testable circuit.

III. 테스트 방법의 구현

1. 테스트 패턴 생성 알고리즘의 구현

무고정 부분 스캔 테스트 방법의 경우 많은 플립-플롭이 스캔되어도 스캔된 회로는 순차 회로 특성을 보유하므로 많은 프레임을 사용하는 것이 어렵다. 따라서 본 연구에서는 주어진 회로를 2나지 3의 짧은 프레임으로 풀어서 이에 대한 반복 배열 표현의 회로를 구성하였다. 여기서 풀린 프레임 수를 k라 한다. 이렇게 풀린 회로가 만들어지면 순차 회로 ATPG 문

제는 조합 회로 순차 회로 문제와 매우 유사하게 된다. 본 연구에서는 풀린 회로에 FAN 알고리즘을 적용하여 테스트 패턴을 구하였다^[3].

FAN 알고리즘은 원래 조합 회로 (combinational circuit)의 결함을 검출하기 위한 테스트 생성 알고리즘으로 그 이전까지 가장 우수한 ATPG로 알려진 PODEM^[2]을 개선한 것으로 PODEM과 유사하게 주로 주입력에서 의사 결정을 함으로서 테스트 벡터를 탐색하는 알고리즘이다. 다만 FAN은 주입력에서 출력 쪽으로 전진한 헤드라인 신호 (head line signal)와 결함 효과가 전달되지 않는 fanout 점에서도 의사 결정을 한다. FAN 알고리즘의 동작을 행위적으로 설명하면 결함 원점에서 결함을 활성화시키기 위하여 주입력, 헤드라인 신호, 결함 효과가 미치지 않는 fanout 점등에 논리 값 조합을 조직적으로 인가하여 한 후 활성화된 결함을 주출력으로 전달하기 위하여 이들 의사 결정점에 논리값을 조직적으로 인가하여 간다. FAN은 이 이외에도 다중 backtrace, X 경로 검사, 유일 감지화 (unique sensitization)등 다양한 기법을 사용하여 PODEM 알고리즘을 개선하였다. 따라서 PODEM 알고리즘에 비하여 그 성능이 월등히 우수하나 그 구현이 까다롭다. 따라서 순차 회로 ATPG, 테스트 패턴 압축 등의 연구를 수행하는 경우 PODEM 알고리즘을 주로 사용하여 왔다. 상용 알고리즘의 경우 FAN이나 FAN이 진화한 SOCRATES 등을 사용하고 있다고 추측되나 자세한 사항은 잘 알려져 있지 않다. FAN 알고리즘은 참고 문헌 [3]에 3개의 미묘한 순서 흐름도로 표현되어 있다. 3개중 2개는 backtrace 절차와 최종 목적 (final objective)결정 흐름도를 각각 나타내는데 에 이를 충실하게 구현하는 것은 어렵지는 않으나 매우 주의를 요하는 작업이다. 본 연구에서는 이 흐름도를 구조화 프로그램 (Structured program) 변경하여 사용하였다.

FAN 알고리즘은 PODEM 알고리즘보다 훨씬 우수하나 순차회로 ATPG에 적용하기가 매우 어렵다. 우리의 ATPG 문제는 프레임 수만큼의 결함이 존재하므로 순차 회로 ATPG 특성을 가지고 있다. 본 연구를 위한 ATPG 알고리즘 개발에서는 특히 다음의 두 가지 사실은 신중히 고려되어야 했다.

사항 1. 결함이 k 프레임에 복수 개 존재한다.

사항 2. 스캔되지 않은 플립-플롭의 첫 번째 프레임

의 논리값은 미지 논리치이다.

이와 유사한 방법이 FASTEST 순차 회로 ATPG에서 적용되었다^[14]. 다만 FASTEST의 경우 모든 플립-플롭의 초기치는 미지 논리값을 가진다. 또한 FASTEST에서는 9-V PODEM이 주 ATPG 알고리즘으로 사용되었다. 우리의 테스트 생성 문제를 위한 ATPG 알고리즘을 구현하는 데에는 9-V PODEM을 적용하는 것이 더 용이하나 본 연구에서는 더욱 진보된 FAN 알고리즘을 사용하기로 하였다. 본 연구에서 FAN 알고리즘을 사용한 이유는 FAN 알고리즘은 SOCRATES 등과 같은 최신 알고리즘으로의 진화가 용이하기 때문이다. 이는 대규모 VLSI 회로 설계에 적용할 수 있고 우수한 테스트 압축 기법을 사용하는 조합 회로 ATPG 기법을 이용한 고성능 ATPG 개발이 가능하게 한다. 그림 12에 테스트 패턴 생성 알고리즘의 최상위 단계 흐름을 보이는 의사 코드가 주어졌다. 그림 12에서 target_fault(i)는 FAN 알고리즘의 결함 원점을 각 프레임에 가하면서 테스트 패턴을 생성함을 보이고 있다. EFAN 함수는 FAN 알고리즘을 무고정 부분 스캔에 적합하게 변형한 FAN의 구현이다.

```

read circuit;
build k-frame iterative array representation;
remaining_faults = all_faults;
while (target_fault =
    select_fault(remaining_faults)) {
    assign U values to uncanned flip-flops
    for(i=0; i<FRAME; i++) {
        test sequence =
            EFAN (target_fault(i), FRAME);
        if(fault is detected) break;
    }
    fill unassigned bits(test sequence);
    fault_simulation(test sequence);
}

```

그림 12. ATPG의 최상위 수준 제어 흐름도
Fig. 12. Top level control flow of ATPG.

2. EFAN 알고리즘 구현

EFAN ATPG 알고리즘의 상위 수준 제어 구조는 원래의 FAN 알고리즘과 동일하다. 다만 테스트 생성을 시작하는 결함 원점이 복수인 점이 다르다. 또한 구성 C 함수들도 복수 결함의 존재와 첫 번째 프레임의 절대적인 제어 불가능성을 고려하여 다르게 구현되어야 한다. 다음에서는 EFAN 알고리즘을 위한 이들

함수의 구현을 FAN과 비교하여 논한다. FAN에는 여러 가지 구성 함수가 있으나 결정적인 계산 과정을 담당하는 함수들과 탐색적인 계산 과정을 담당하는 함수들 두 가지 그룹으로 구분 할 수 있다. 결정적인 계산 과정은 implication 함수가 주로 담당하며 탐색적인 계산 과정은 backtrace와 backtrack을 담당한다. 유일 감지화 (unique sensitization) 과정 등은 결정적 계산 영역을 최대한 확장하는 역할을 한다. 아래에서는 이들 세 가지 계산 과정에 대하여 상세히 논한다. 이들을 조합하여 완전한 ATPG 알고리즘을 완성하는 EFAN의 흐름도는 FAN과 유사하므로 구현시 참고 문헌 [3]의 FAN 흐름도를 참조하면 된다.

Implication 구현

implication 함수는 EFAN 알고리즘에서 가장 많은 시간을 소모하는 과정으로 효과적으로 구현되어야 한다. implication 함수의 구현에서 가장 중요한 문제는 논리치 시스템을 정하는 것이다. FAN 알고리즘은 조합회로 전용 ATPG 알고리즘이다. 조합회로에서는 본 논문의 경우와 달리 오직 하나의 결함원 (fault origin)이 있다. FAN의 저자들은 유일 결함원 조건을 최대한 이용하였다. FAN은 D-알고리즘과는 달리 테스트 패턴을 탐색하는 과정에서 의사 결정을 오직 결함 효과가 전달되지 않는 점에서 수행한다. 결함 효과가 전달되지 않는 점은 원래 회로와 결함 회로가 동일한 논리치를 가진다. 이로 인하여 5 논리치 시스템을 사용하여도 쉽게 ATPG 알고리즘의 완전성을 보장할 수 있다. 일단 알고리즘의 완전성이 보장되면 알고리즘의 성능이 중요하게 된다. 이 경우 간단하지만 전통적인 컴퓨터 구조에서 쉽게 구현될 수 있는 기법이 강력하지만 구현이 복잡한 기법보다 유리한 경우가 많다. 5 논리치 시스템은 9 논리치 시스템과 같이 회로의 상태를 세밀하게 유지할 수 없으나 구현이 매우 용이한 장점이 있다. PODEM과 같이 완전히 전진 implication에 의존하는 경우에는 9 논리치 시스템을 사용하는 것이 전혀 문제가 되지 않는다. 그러나 FAN과 같이 후진 implication에 많이 의존하는 알고리즘의 경우 논리값의 종류가 많아지면 효과적인 구현이 어려워진다. 따라서 FAN 알고리즘의 경우 5 논리치 시스템을 사용하는 것은 매우 적절한 결정이다.

무고정 부분 스캔 테스트 방법을 위한 ATPG 알고리즘은 FAN을 확장한 것이나 다른 어느 순차회로

ATPG의 경우에서처럼 적어도 9 논리 시스템이 사용되어야 한다. 그 이유는 5 논리치 시스템을 사용하는 경우 완전성을 보장할 수 없기 때문이다. 예를 들어 5 논리치를 사용하면 그림 7의 b-s-1 결함은 플립-플롭 A만 스캔한 경우 검출 패턴 생성이 불가능하게 된다. 본 연구에서는 스캔되지 않는 플립-플롭을 U로 초기화하여 원래 회로 (good machine)와 결함 회로 (faulty machine)는 각각 0, 1, X, U의 네 가지 논리값을 가져 결함 회로 (combined machine)는 16개의 가능한 논리치를 가지는 16 논리치 시스템을 사용하였다. 새로운 U 값은 X 값과는 달리 논리치 0 이나 1로 바뀔 수 없는 값으로 테스트 패턴 탐색에 유효하게 사용된다.

표 1 에는 AND, OR, NOT 연산자의 implication 표가 주어져 있다.

표 1. Implication 표
Table 1. Implication table.

	AND				OR				NOT
	0	1	X	U	0	1	X	U	
0	0	0	0	0	0	1	X	U	1
1	0	1	X	U	1	1	1	1	0
X	0	X	X	X	X	1	X	X	X
U	0	U	X	U	U	1	X	U	U

EFAN의 implication 구현이 FAN의 경우보다 어려운 점은 후진 implication의 구현에 있다. FAN의 경우 후진 implication은 항상 결함 효과가 전달되지 않는 영역에서 이루어진다. 따라서 후진 implication에서는 5 논리치 중 0, 1의 두 가지 값만을 신호선에 가하게 된다. EFAN의 경우 결함 효과가 전달되는 영역이 매우 크므로 후진 implication을 결함 효과가 전달되는 영역까지 확대할 필요가 있다. 이 필요성은 확장된 유일 감지화 (unique sensitization) 기능에 의한 것으로 다음에 설명한다. 본 연구에서는 imply 함수는 원래 회로와 결함 회로에서 독립적으로 수행된다. 전진 implication은 두 회로를 합친 결함 회로에서 수행되게 구현하는 것이 용이하나 후진 implication은 각각 따로 구현하여야 하기 때문이다.

16 논리치 시스템은 implication때 사용되는 사건 시뮬레이션의 구현에도 영향을 미친다. 즉 FAN의 경우 사건의 중복을 방지하는 flag와 어떤 신호의 논리치가 완전히 결정되었음을 말하는 flag가 공유되어 매

우 효과적으로 구현될 수 있다. 16 논리치를 사용하면 1/X, X/0 등과 같이 부분 적으로 논리치가 정해지는 경우가 있기 때문에 논리치 계산이 완료된 상태를 나타내는 flag가 추가적으로 필요하다. 현재까지의 연구 결과 EFAN의 imply 알고리즘은 FAN의 imply 알고리즘보다 매우 계산량이 많음을 발견하였다. 먼저 EFAN의 implication은 FAN의 경우 보다 매우 세밀한 논리 상태를 유지하여야 한다. 세밀한 논리 상태 유지는 ATPG 알고리즘의 성능을 위한 것이 아니라 완전성을 위한 것이기 때문에 피할 수 없다. 또한 후진 implication 루틴의 구현이 다소 복잡하여 일반적인 컴퓨터 구조상에서 효과적으로 프로그램하기 어렵다. 이 문제는 ATPG를 위한 특수 명령을 가진 컴퓨터를 사용하여 해결할 수 있으나 비현실적인 해결책이다.

Backtrace와 Backtrack

FAN에 기반을 둔 ATPG 알고리즘의 탐색적 계산 과정은 backtrace와 backtrack으로 이루어진다. backtrace는 탐색을 위한 의사 결정을 하는 논리 신호선과 그 논리치를 결정하는 과정이며 backtrack은 의사 결정이 잘못 되었음을 발견하였을 때 그 의사 결정이 이루어지지 않은 상태로 복구하는 과정이다. 탐색 의사 결정 알고리즘에 의하여 어떤 신호선에 어떤 논리값이 결정되면 위에서 논한 implication 과정에 의하여 의사 결정 효과를 전파한다. backtrace 과정의 입력은 목적 리스트 (objective list)가 된다. FAN 알고리즘에서는 어떤 논리 연산자 (logic gate)의 입력단에 존재하는 결합 효과를 출력으로 전달하기 위해서는 그 연산자의 다른 입력 신호는 1/1 이나 0/0 가 그 연산자가 AND이나 OR에 따라 각각 인가되어야 한다. EFAN의 경우에는 다른 입력 신호들은 전달된 결합 효과의 극성과 연산자의 종류에 따라 1/X, 0/X, X/0, X/1 등으로 인가되어야 한다. 또한 원래 회로와 결합 회로가 동일한 논리값을 가진다는 확신이 선 경우에는 FAN 알고리즘에서처럼 0/0나 1/1을 인가하여야 한다. 이렇게 결정된 게이트와 인가할 논리치는 implication 과정 수행 결과 얻어진 정당화되지 않은 모든 게이트들과 합하여 초기 목적 리스트 (initial objective list)를 구성한다.

Backtrace 과정은 이 목적 리스트를 처리하여 의사 결정점과 결정 논리값을 계산한다. EFAN에서는

FAN의 경우와 마찬가지로 비록 imply 함수에서는 16 논리값을 사용하여 세밀한 회로 상태를 기록하여야 하나 주입력이나 fanout 점에서의 논리값을 결정하여야 할 경우에는 원래 회로와 결합 회로가 동시에 같은 값으로 결정한다. 즉, 논리값이 0/X, X/0 가 요구될 경우에는 0/0 로 결정되며 1/X, X/1 가 요구될 경우에는 1/1로 결정된다. FAN에서 마찬가지로 결합 효과가 fanout 점에 전달되지 않을 경우에만 fanout 점은 의사 결정점이 된다. 초기 목적 리스트의 논리 값 요구는 원래 회로 (good machine), 결합 회로 (faulty machine), 결합 회로 (combined circuit)등 모두에 대하여 가능하다. 결합 효과를 전파하기 위한 초기 목적(initial objective)과 유일 감지화 과정의 결과 발생한 초기 목적은 원래 회로 혹은 결합 회로 값만 요구하거나 두 회로 모두의 값을 요구할 경우에도 동일한 논리치를 요구한다. 그러나 implication의 과정에서는 1/0 이나 0/1과 같이 두 회로가 서로 다른 값을 요구하는 경우가 있을 수 있다. backtrace 과정은 탐색을 위한 heuristic으로 ATPG 알고리즘의 완전성에는 영향을 주지는 않으나 성능에 영향을 미칠 수 있다. 따라서 가장 효과적인 backtrace 구현에는 많은 알고리즘 및 프로그래밍 관련 결정을 내려야 한다. 이들 결정은 오직 실험적으로 그 효과가 검증되어야 한다. 본 연구에서는 FAN의 다중 backtrace를 변경하여 사용하였다. FAN의 다중 backtrace는 결합이 전파되지 않는 fanout 신호선에서 backtrack을 중지하는 점 이외에 결합 효과에 의하여 전혀 영향을 받지 않는다. EFAN 알고리즘의 경우 다중 결합이 존재하므로 결합 효과가 전파되는 영역이 커지는 점을 고려하여 결합 효과에 대한 고려를 증가하였다. 이는 초기 목적 요구가 0/X, 1/X, X/0, X/1처럼 원래 회로와 결합 회로에 독자적으로 나타나는 경우를 효과적으로 고려하여야 하기 때문이다. 그러나 결합 회로의 구조는 원래 회로의 구조와 거의 동일하므로 많은 결합 회로 목적 요구는 주입력 값에 의하여 만족되어야 한다. 이러한 이유로 인하여 FAN의 다중 backtrace에서 사용되는 각 논리치 0, 1이 각각 요구되는 횟수를 나타내는 변수는 두 회로를 통합하여 하나의 변수로 사용하였다. 그러나 원래 회로나 결합 회로의 신호선에 이미 원하는 논리치가 이미 가하여져 있는 경우 무의미하게 backtrack 하는 것을 방지하기 위하여 backtrack이 원래 회로 상에서인지 결합 회로 상인지

두 회로 모두인지를 나타내는 GOOD, FAULTY, PAIR를 나타내는 변수 fg_expect 를 첨가하였다. 원래 회로와 결함 회로가 서로 다른 값을 요구할 경우 논리치 0 과 1 의 요구 수를 각각 1, 1로 나타내거나 원래 회로에 우선권을 주어 1, 0 로 나타낼 수 있다. 전자의 경우는 fg_expect 를 PAIR로 하고 후자의 경우는 fg_expect 는 GOOD로 하게 된다. 본 논문에서는 원래 회로에 우선권을 주는 방식을 사용하였다. fanout 점에서 원래 회로와 결함 회로가 서로 다른 논리치를 요구할 경우에는 fg_expect 를 PAIR로 하고 0 와 1 의 요구되는 수를 더하여야 한다. 이 경우 결함 회로 요구와 원래 회로 요구가 합쳐져 정확성을 잃게 된다. 이 사실이 ATPG 성능에 미치는 영향이 크지 않을 것으로 생각되나 후후 자세히 조사할 예정이다. 그림 13 에는 본 연구에서 사용한 backtrace 알고리즘 적용례가 주어져 있다.

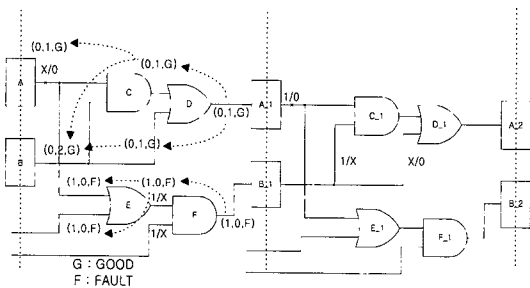


그림 13. 다중 backtrace의 예
Fig. 13. Example for multiple backtrace.

그림 13 에서는 원래 회로와 결함 회로에 유일 감지화에서 각각의 회로에 논리치가 인가됨에 따라 원래 회로와 결함 회로에 각기 다른 목적 요구가 생기는 경우를 보이고 있다. 즉 유일 감지화에 의해 결정된 논리치를 통해 F 연산자는 1/0의 원래 회로와 결함 회로에 각기 다른 목적 요구가 생기게 된다. 이미 결정된 회로의 논리치에 대해서는 다중 backtrace를 하지 않으며, 또한 결함이 전파되지 않는 연산자에 대해서는 제어가 가장 용이한 라인이라 할지라도 원래 회로와 결함 회로가 각기 다른 논리치로 다중 backtrace를 하지 않는다. 따라서 F 연산자는 E 연산자 방향으로 결함 회로에 대해서만 다중 backtrace를 하게 된다.

다중 backtrace에 의하여 가장 먼저 결정하여야 할 신호선 값이 결정되면 해당 신호선에 가한 후 im-

plication 절차를 수행한 후 결함이 주출력으로 검출되고 모든 정당화 요구(justification requirements)가 만족되었는지를 확인한다. 이 경우 테스트 패턴 생성이 완료되며 그렇지 않은 경우 이러한 과정을 반복한다. implication 결과 충돌 (conflict) 이 생기거나 D 프론티어가 사라지는 경우 backtrack을 수행한다. backtrack은 이전의 잘못된 의사 결정의 결과 테스트 패턴 탐색을 계속할 수 없는 경우에 의사 결정을 반복하는 과정이다. backtrack을 정확하게 구현하기 위해서는 의사 결정 이전 상태로 회로의 상태를 되돌려 놓고 새로운 의사 결정을 할 수 있게 하여야 한다. FAN의 경우 5 논리치 시스템을 사용하므로 탐색 과정에서 논리치는 모두 X에서 0, 1, D, \bar{D} 로 변화하게 된다. 따라서 회로의 상태를 되돌려 놓기 위해서는 의사 결정에 의하여 변경된 게이트들만 의사 결정 스택에 넣으면 된다. EFAN의 경우 16 논리치를 사용하므로 변경 게이트와 변경 회로 쌍을 스택에 넣던지 변경 게이트와 변경 전 논리치 쌍을 스택에 넣으면 된다. 이러한 점 또한 EFAN이 FAN 보다 계산적으로 복잡함을 보이고 있다.

유일 감지화 (Unique Sensitization) 구현

유일 감지화를 위하여 먼저 전처리 과정에서 모든 결함 원점에서 모든 프레임의 모든 연산자까지의 결함 전파 극성을 먼저 계산한다. 여기서 세 가지의 경우를 얻을 수 있다. 결함 전파가 되지 않는 연산자에 대해서는 원래의 FAN과 마찬가지로 AND, OR 연산자에 따라 1/1, 0/0의 값을 결함 경로가 아닌 입력에 인가하여 결함 효과를 전진시킨다. 결함 효과가 전파될 가능성이 있는 연산자는 그 전파 극성을 전처리 과정에서 계산한다. 전파 극성은 모든 프레임의 결함원으로부터 어떤 게이트에 이르는 모든 경로의 결함 전파 극성을 조사하여 결정하는데 전파 극성에는 POS, NEG, ERR가 있다. 여기서 ERR 극성은 결함 전파 극성의 충돌이 있음을 나타낸다. 유일 감지화를 위한 목적 요구는 결함 전파 극성과 연산자의 종류에 따라 1/X, 0/X, X/1, X/0의 값이 결정되게 된다. ERR 극성을 가진 연산자에 대해서는 ATPG 과정에서 현재의 D 프론티어에서 다시 전파 극성을 계산하게 된다. 이를 동적 극성 계산이라 하는데 테스트 패턴 탐색 과정에서 가하여진 신호들의 논리치에 의하여 결함 전파가 중단되는 경로를 고려하여 결함 전파 극성을 다시 계

산하는 과정이다. 동적으로 계산된 극성은 backtrack 과정에서 다시 원래의 극성으로 복구되어야 한다. 다음은 동적으로 극성이 결정되는 예이다.

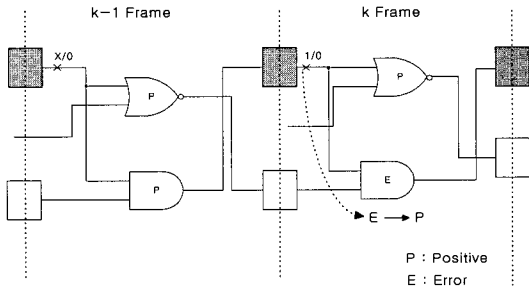


그림 14. 동적 결함 전파 극성 결정
Fig. 14. Determining dynamic fault effect propagation polarity.

그림 14에서는 k-1 프레임에서의 음의 결함 전파 극성과 k 프레임에서의 양의 결함 전파 극성으로 인해 충돌이 발생한 연산자가 k 프레임에서 시작된 유일 감지화로 인하여 전파 극성이 양으로 결정이 된다. 따라서 논리치는 양의 전파 극성과 AND 연산자에 의해 1/X라는 값이 결정이 된다. 결정된 1/X는 implication을 통해 k-1 frame에서 결함 신호선에 0/X의 값을 요구함으로써 원래 결함에 의해 이미 결정된 X/0의 논리치와 충돌없이 인가 가능하게 된다.

EFAN에서 16-논리치를 사용하기 때문에 5-논리치를 사용하는 것에 비해 값을 결정하여 연산자에 인가하는 단계에서도 더 복잡한 연산 과정이 수행된다. 이미 연산자에 인가된 값이 인가하려는 값의 포함 여부에 따라 그 값의 인가 가능 여부를 결정하며 가능시 이미 인가된 값과 인가하려는 값을 합치는 과정이 필요하다. 예를 들어 5-논리치에서는 1의 값을 인가하기 위해서는 인가 라인의 논리치가 1 또는 X이면 인가가 가능하나, 16-논리치에서는 1/X를 인가하기 위해서 인가 라인의 논리치가 1/X, 1/0, 1/1, 1/U, X/X, X/1, X/0, X/U 중의 하나임을 확인하고 원래의 논리치와 인가 라인의 논리치를 합쳐서 1/X, 1/0, 1/1, 1/U 중의 한 가지 형태값으로 인가 라인의 논리치가 변하게 된다.

FAN에서 사용되는 dominator개념은 EFAN에서도 유효하나 결함 전파 경로들의 극성이 다른 경우 동적으로 일부 경로가 막히는 지를 검사하여야 한다. 이 검사 과정은 매우 복잡하여 본 연구에서는 구현되지

않았다. 이에 대한 연구 결과는 추후 다른 곳에서 보고할 예정이다.

3. EFAN 알고리즘의 효율성

먼저 국지적이지 않는 케환이 없는 회로는 주어진 회로를 최대 순차 깊이 (maximal sequential depth) 만큼 풀면 효과적으로 ATPG 할 수 있을 것으로 기대된다. 그러나 자체 케환 (self loop)이 없고 그래프 이론을 이용하여 원거리 케환을 없애고 아울러 최대 순차 깊이를 아주 작게한 부분 스캔 회로의 경우에도 제안된 ATPG 알고리즘으로 테스트를 구할 수 없는 경우가 있다. 다음에서는 이러한 경우에 스캔하지 않아도 완전 스캔 결함 검출율을 거의 감소시키지 않는 최대 플립-플롭의 집합을 구하는 알고리즘을 논한다.

4. 부분 스캔 선택 절차

부분 스캔 테스트 방법에서는 ATPG 알고리즘이 효과적이지 못할 경우에는 필요한 결함 검출율을 얻기 위하여 최소한의 플립-플롭을 스캔하여야 한다. 먼저 비국지적인 케환 구조는 대부분의 경우 순차회로 ATPG를 어렵게 하는 경향이 있으므로 참고 문헌 [18]의 알고리즘을 이용하여 회로 구조를 케환 구조가 없고 최대 순차 깊이가 적게 되는 플립-플롭의 집합을 구한다. [18]에서 제안된 방법은 먼저 주어진 그래프에 관련된 S-그래프를 구한다. S-그래프는 순차 회로의 순차 깊이와 케환 구조를 나타내는 추상적인 그래프이다.

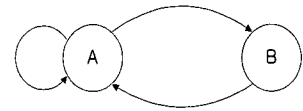


그림 15. 그림 7 회로의 S-그래프
Fig. 15. The S-graph of the circuit in Fig 7.

그림 15 에는 그림 7 회로의 S-그래프가 주어져 있다.

S-그래프의 노드 (node)는 플립-플롭을 나타내며 어떤 플립-플롭에서 다른 플립-플롭으로 조합 경로가 있으면 에지 (edge) 가 존재한다. 스캔되는 플립-플롭의 수는 면적 부담과 직접적인 관계가 있으므로 면적 부담을 줄이기 위하여 최소한의 플립-플롭이 선택되어야 한다. S-graph가 케환이 없게 하는 플립-플롭을 선택한 후 회로의 순차 깊이를 줄인다. 순차 깊이가

매우 작은 경우가 대부분이므로 경로 길이가 $d+1$ 이상인 모든 경로를 나열한 후 커버링 알고리즘을 이용하여 최소한의 노드를 선택하여 경로 길이를 d 이하로 만들 수 있다. 참고 문헌 [18]에는 전술한 두 문제를 효과적으로 푸는 알고리즘이 기술되어 있다.

어떤 회로의 경우, S-그래프가 사이클이 없고 순차 깊이가 프레임 수보다도 작은 경우에도 사용된 ATPG 알고리즘이 높은 검출율을 가진 테스트를 생성하는 데에 효과적이지 못한 경우도 있다. 예를 들어 그림 15의 S-그래프에서 노드 A에 해당하는 플립-플롭을 스캔하여도 결함 검출율이 높지 않음을 앞에서 보인 바 있다. 다음에는 ATPG에 기반을 둔 부분 스캔 선택 알고리즘을 논한다. 사실 아래 알고리즘에서는 결함 검출율을 감소시키지 않는 최대한의 비스캔 플립-플롭의 집합을 구한다. 사용된 알고리즘은 다음과 같이 동작한다. 먼저 주입력과 주출력, 그리고 위에 기술한 그래프 이론 기반의 부분 스캔 알고리즘으로 선택된 플립-플롭에 작은 제어도 수와 관측도 수를 할당한다. 나머지 플립-플롭에는 큰 제어도 수와 관측도 수를 할당한다. 여기서 큰 수는 낮은 제어도나 관측도를 나타내면 작은 수는 높은 제어도나 관측도를 나타낸다. 다음 이들을 이용하여 전체 풀린 회로의 제어도와 관측도를 계산하고 모든 결함에 대하여 ATPG를 수행한다.

ATPG 수행 도중 각 플립-플롭마다 플립-플롭에 어떤 값이 정해진 수와 그 플립-플롭을 이용하여 결함에 대한 테스트가 발견된 횟수를 기록한다. 가장 적게 사용된 플립-플롭이 비스캔 대상 플립-플롭이다. 어떤 플립-플롭이 관측에 사용된 횟수는 값이 정해진 수보다 큰 가중치를 두어야 한다.

IV. 실험 결과

제안된 부분 스캔 테스트 방법의 효율성을 검증하기 위하여 III 절의 알고리즘을 C 언어로 구현하였다. 결함 검출율을 구하기 위하여 간단한 유일 결함 전파 결함 시뮬레이터를 구현하였다. 임의 패턴은 전혀 사용하지 않았으나 테스트 시퀀스 탐색 완료 후 남은 비트들은 임의로 정하였다.

테스트 시퀀스가 언어지면 결함 시뮬레이터를 이용하여 그 테스트 시퀀스에 의하여 검출되는 모든 결함을 결함 리스트에서 제거하였다.

표 2. 회로 S5378에서의 결함 검출율 변화
Table 2. Fault Coverage Change in S5378.

F/Fs	0	10	20	30	40	50
Cov.	99.19	98.87	98.47	98.29	98.25	98.05
F/Fs	60	70	80	90	100	135
Cov.	96.58	93.94	92.17	84.56	72.85	51.85

표 2에서는 S5378에서 비스캔 플립-플롭이 증가할 경우의 결함 검출율 감소를 보이고 있다. 약 60개의 플립-플롭이 스캔되지 않았을 경우 결함 검출율이 급격히 떨어짐을 알 수 있다.

표 3. 비스캔 플립-플롭을 포함한 경우의 결함 검출율

Table 3. Fault coverage with unscan flip-flops.

Circuit	Total F/Fs	Org. Fault Cov. (%)	Self Loop	unscan F/Fs	unscan ratio	Fault Cov. (%)
S27	3	100	3	2	66.7	100
S298	14	100	14	2	14.3	93.57
S382	21	100	15	4	19.0	95.32
S444	21	97.34	15	3	14.3	90.11
S526	21	99.85	21	4	19.0	96.84
S838	32	100	32	1	3.1	91.74
S1196	17	100	0	5	29.4	99.76
S1238	18	95.21	0	6	33.3	95.07
S5378	179	99.19	0	42	23.5	98.17
S9234	228	93.79	150	28	12.3	92.47

표 3에서는 나타난 수만큼의 플립-플롭이 스캔되지 않았을 경우의 결함 검출율을 보였다. 세 번째 난에는 본 연구를 위하여 구현한 FAN 알고리즘 사용할 경우의 결함 검출율을 나타낸다. 다섯 번째 난에는 스캔되지 않은 플립-플롭의 수를 나타낸다. 여섯 번째 난에는 스캔되지 않은 플립-플롭의 비율을 나타내며 일곱 번째 난에는 각각 경우의 결함 검출율을 나타낸다. 결함 검출율이 갑자기 떨어지는 점에서 비스캔을 중단하였다. 자체 궤환 플립-플롭이 적은 경우에는 제안된 부분 스캔 선택 방법이 결함 검출율의 현저한 감소 없이 20% 이상의 플립-플롭을 스캔하지 않을 수 있음을 보였다. 실험 도중 자체 궤환이 많은 회로의 경우 제안된 부분 스캔 방법이 비효율적임이 발견되었다. 예를 들어 S838 회로의 경우 오직 하나의 플립-

플롭이 스캔되지 않을 경우에도 결합 검출율이 100%에서 91.74%로 급격히 떨어졌다. 사실상 S838 회로는 초기화 문제로 인하여 어떠한 순차회로 ATPG 알고리즘도 처리하기 어렵다. 그러나 스캔되지 않는 플립-플롭을 논리값 1로 초기화한 경우 100%의 결합 검출율을 얻을 수 있었다.

표 4. 다른 부분 스캔과의 결과 비교
Table 4. Comparison of results for other partial scan method.

Circuit	SCAN F/Fs (OPUS)	Fault Cov. (%)	Vec. Count (HITEC)	SCAN F/Fs	Fault Cov (%)	Vec. Count (EFAN)
S298	6	99.35	103	12	93.57	30
S444	9	95.99	184	18	90.11	42
S526	9	98.92	369	17	97.59	69
S1196	6	100	400	12	99.78	145
S1238	6	94.91	394	12	95.07	154
S5378	69	97.20	795	137	98.17	296
S9234	100	37.60	770	200	92.47	407

표 4는 일반적으로 사용되는 부분 스캔 기법을 이용하여 HITEC을 통해 테스트 벡터를 생성한 결과와 제안된 부분 스캔 방법과 EFAN을 사용하여 테스트 벡터를 생성한 결과를 비교한 것이다. 일반적인 부분 스캔 방식은 원래 스캔된 플립-플롭이외의 플립-플롭에 스캔 동작시 플립-플롭의 데이터를 고정하는 방법이 필요하게 된다. 상용 툴에서 클럭을 게이팅하거나 스캔 플립-플롭을 hold하는 등의 많은 방식을 제시하고 있으나, 여기서는 Syntest에서 제공하는 방식^[32]을 이용해서 부분 스캔을 하였다. 이 때 스캔된 플립-플롭의 수는 각각의 부분 스캔 기법에 대해서 같은 면적 부담을 가지는 것을 기준으로 한다. 두 번째 난은 일반적인 부분 스캔 기법인 OPUS를 이용해서 같은 면적 부담을 가지는 만큼의 플립-플롭을 강제적으로 선택한 것이고 세 번째 난과 네 번째 난은 순차회로 ATPG를 통해서 테스트 결과를 나타낸 것이다. 여섯 번째 난과 일곱 번째 난에서 보듯이 벡터의 수가 일반적인 순차회로 ATPG에 비해 매우 적은 반면 결합 검출율은 비슷함을 알 수 있다.

V. 결론 및 추후 연구 과제

본 논문에서 우리는 흥미있는 부분 스캔 테스트 방

법을 소개하였다. 제안된 방법은 스캔 입출력시 스캔되지 않은 플립-플롭에 데이터 고정 기능이 없다. 따라서 완전 스캔 설계 기법과 테스트 목적 설계 변형 요구나 테스트 벡터 적용 방법이 동일하다. 따라서 클럭 게이팅이나 복수 클럭 나무 구조를 제공할 필요가 없다. 또한 높은 결합 검출율을 얻을 수 있고 테스트 압축도 용이한 조합 회로 ATPG 알고리즘인 FAN에 기반한 EFAN ATPG 알고리즘을 사용하였다. 앞으로 EFAN 알고리즘을 개선하여 성능을 높이고 테스트 압축 기법에 관한 연구를 완료하면 대규모 VLSI 회로를 테스트할 수 있는 부분 스캔 테스트 방법을 얻을 수 있을 것으로 기대 된다. 이를 위하여 우리는 다음과 같은 연구를 수행할 예정이다.

먼저 현재의 ATPG 구현은 다소 불완전하여 완전 스캔 ATPG를 수행할 경우 원래의 FAN 알고리즘보다 다소 낮은 성능을 보이는 경우가 있다. 이는 주로 유일 감지화 (unique sensitization) 구현상의 불완전성으로 보인다. 현재 완전하며 속도가 빠른 EFAN 구현이 진행 중이다. 또한 SOCRATES등에서 사용된 학습에 기반을 둔 기법들을 이용하여 EFAN의 성능을 증가시키는 연구를 진행할 예정이다.

제안된 부분 스캔 방법은 많은 개선의 여지가 있다. 특히 자체 궤환 플립-플롭을 다수 포함한 회로에 대한 테스트 목적 설계 변경 및 ATPG 방법에 대한 연구를 수행할 예정이다. 우리는 모든 자체 궤환 구조가 ATPG에 어려움을 준다고 믿지 않는다. 따라서 ATPG에 어려움을 주지 않는 자체 궤환 구조를 연구하여 그렇지 않는 자체 궤환 구조를 테스트 목적상 변경하는 방안도 연구하고 있다. 특히 자체 궤환 구조는 데이터 연산부보다 회로의 제어 부분에 많을 것으로 보인다. 이러한 경우 회로의 기능상 자체 궤환을 가진 플립-플롭에 비동기 set/reset 신호를 포함하는 경우가 많다. 이 경우 비동기 set/reset 기능을 이용하는 방법을 연구하는 것이 중요하다. 상기의 연구 결과와 더불어 본 연구를 시작하게 된 중요한 모티브인 높은 결합 검출율을 가지면서 매우 압축된 테스트 시퀀스를 생성하는 방법에 관한 연구도 진행하고 있다.

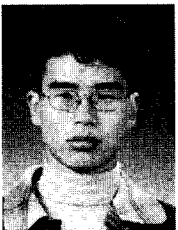
참 고 문 헌

[1] J. P. Roth, "Diagnosis of Automata Failures: A Calculus and A Method,"

- IBM J. Res. Develop., vol. 10, pp. 278-291, July 1966.
- [2] P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits," *IEEE Trans. Comput.*, vol. C-30, pp. 215-222, Mar. 1981.
- [3] H. Fujiwara and T. Shimono, "On the Acceleration of Test Generation Algorithms," *IEEE Transactions on Computers*, vol. c-32, no. 12, December 1983, pp. 1137-1144.
- [4] M. H. Schulz, E. Trischler, and T. M. Sarfert, "SOCRATES: A Highly Efficient Automatic Test Pattern Generation System," *IEEE Transactions on Computer-Aided Design*, vol. 7, no. 1, January, pp. 126-136.
- [5] M. H. Schulz and E. Auth, "Improved Deterministic Test Pattern Generation with Application to Redundancy Identification," *IEEE Transactions on Computer-Aided Design*, vol. 8, no. 7, July 1989, pp. 811-816.
- [6] W. Kunz and D. K. Pradhan, "Accelerated Dynamic Learning for Test Pattern Generation in Combinational Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 5, May 1993, pp. 684-693.
- [7] G. Tromp, "Minimal Test Sets for Combinational Circuits," *ITC-1991*, pp. 204-209.
- [8] S. Kajihara, I. Pomeranz, K. Kinoshita, and S.M. Reddy, "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits," *TCAD*, vol. 14, no. 12, Dec. 1995, pp. 1496-1503.
- [9] I. Pomeranz, L.N. Reddy, and S.M. Reddy, "COMPACTEST: A Method to Generate Compact Test Sets for Combinational Circuits," *ITC-1991*, pp. 194-203.
- [10] J.S. Chang and C.S. Lin, "Test Set Compaction for Combinational Circuits," *TCAD*, vol. 14, no. 11, Nov. 1995, pp. 1370-1378.
- [11] T.M. Niermann and J.H. Patel, "HITEC: A Test Generation Package for Sequential Circuits," in *Proc. European Design Automation Conf.*, Feb. 1991.
- [12] W.T. Cheng, "The Back Algorithm for Sequential Test Generation," *ICCAD 1988*, pp. 66-69.
- [13] D.H. Lee and S.M. Reddy, "A New Test Generation Method for Sequential Circuits," *ICCAD-1991*, pp. 446-449.
- [14] T. P. Kelsey, K. K. Saluja, "Fast Test Generation for Sequential Circuits," *ICCAD-1989*, pp. 354-357.
- [15] E. Trischler, "Incomplete Scan Path with an Automatic Test Generation Methodology," *International Test Conference*, 1980, pp. 153-162, Nov. 1980.
- [16] R. Gupta, R. Gupta, and M.A. Breuer, "BALLAST: A Methodology for Partial Scan Design," in *Proc. 19th Symp. on Fault-Tolerant Computing*, pp. 118-125, June 1989.
- [17] K.T. Cheng and V.D. Agrawal, "A Partial Scan Method for Sequential Circuits with Feedback," *IEEE Transactions on Computers*, vol. 39, no. 4, pp. 544-548, Apr. 1990.
- [18] D.H. Lee and S.M. Reddy, "On Determining Scan Flip-flops in Partial Scan Designs," in *Proc. Int. Conf. on Computer-Aided Design*, pp. 322-325, Nov. 1990.
- [19] V. Chikermane and J.H. Patel, "A Fault Oriented Partial Scan Design Approach," *ICCAD-1991*, pp. 400-403.
- [20] M. Abramovici, J.J. Kulikowski, and R.K. Roy, "The Best Flip-flops to Scan," *International Test Conference*, pp. 166-173, Oct. 1991.
- [21] D. Xiang and J.H. Patel, "A Global Algorithm for the Partial Scan Design Problem Using Circuit State Information," *International Test Conference*, pp.

- 548-556, October 1996.
- [22] C. Gloster and F. Brglez, "Partial Scan Selection for User-Specified Fault Coverage," Euro-DAC-95.
- [23] M. Abramovici, P. S. Parikh, B. Mathew, and D. G. Saab, "On Selecting Flip-flops for Partial Reset," International Test Conference, pp. 1008-1012.
- [24] V. Chickermane, E.M. Rudnick, P. Banerjee, and J.H. Patel, "Non-Scan Design-for-Testability Techniques for Sequential Circuits," DAC-93, pp. 236-241.
- [25] I. Park, D.S. Ha, and G. Sim, "A New Method for Partial Scan Design Based on Propagation and Justification Requirements for Faults," International Test Conference, 1995, pp. 413-422.
- [26] I. Hartanto, V. Boppana, and W. K. Fuchs, "Identification of Unsettable Flip-flops for Partial Scan and Faster ATPG," ICCAD-96, pp. 63-66.
- [27] P. Pan and C. L. Liu, "Partial Scan with Pre-selected Scan Signals," DAC-95, pp. 189-194.
- [28] C. Lin, M. Marek-Sadowska, K.T. Cheng, and M.T. Lee, "Test Point Insertion: Scan Paths through Combinational Logic," DAC-96, pp. 268-273.
- [29] T.M. Niermann, R. K. Roy, J. H. Patel, and J.A. Abraham, "Test Compaction in Sequential Circuits," IEEE Transactions on Computer-Aided Design, vol. 11, no. 2, Feb. 1992, pp. 260-267.
- [30] I. Pomeranz and S.M. Reddy, "On Generating Compact Test Sequences for Synchronous Sequential Circuits," Proceedings on European Design Automation Conference, Sept, 1995.
- [31] I. Pomeranz and S.M. Reddy, "On Static Compaction of Test Sequences for Synchronous Sequential Circuits," 33rd DAC, pp. 215-220.
- [32] "Syntest User's manual," version 1.0.9, 1996.

 저 자 소 개



盧 炫 澈(正會員)

1996년 2월 경북대학교 전자공학 학사. 현재 경북대학교 전자공학과 대학원 석사과정 재학중. 주관심분야는 집적 회로 설계 및 테스트, 설계 자동화

李 東 浩(正會員)

1979년 2월 서울대학교 전자공학과 졸업. 1981년 2월 KAIST 전산학과 졸업(이학석사). 1981년 3월 ~ 1992년 7월 ETRI 선임 연구원. 1992년 8월 (미) 아이오와 대학 전산학과 졸업(박사). 1992년 8월 ~ 1993년 1월 (미) Motorola senior CAD engineer. 1993년 3월 ~ 현재 경북대학교 전자공학과 조교수. 주관심분야는 컴퓨터 구조, 설계 자동화, 프로그래밍 언어