

IEEE Std. 1500 인터페이스를 사용한 Programmable MBIST 구조

Programmable MBIST Architecture Using IEEE Std. 1500 Interface

박기현, 김근배, 김일웅, 강일권, 양동훈, 강성호

연세대학교 전기전자 공학부

e-mail : {cimabear, kgb9572, woong, neokings, ysydh}@soc.yonsei.ac.kr,

shkang@yonsei.ac.kr

Abstract

In this paper, a hardware-centric distributed Memory Built-In Self-Test(MBIST) architecture using IEEE standard 1500 interface is proposed. The proposed architecture can concurrently execute distinct memory test algorithms for heterogeneous memories. We compared the proposed MBIST architecture with the existing MBIST architecture and found that the proposed one has precedence of the existing ones.

I. 서론

현재 세계 반도체 산업의 기술환경은 System-on-Chip(SoC) 경향으로 나아가고 있다. 그리고 최근의 SoC 칩에서 내장된 메모리가 트랜지스터와 실리콘 면적 부분에서 가장 큰 비중을 차지하고 있다[1]. 그러므로 내장된 메모리의 테스트는 성공적인 SoC 칩의 설계와 공정을 위해서 매우 중요하다. 그러나 내장된 메모리는 외부에서의 테스트 접근이 용이하지 않다. 그렇기 때문에 SoC에서의 효율적인 내장 메모리 테스트를 위한 BIST의 연구가 활발히 진행되고 있다.

오늘날, SoC안에는 다양한 종류의 내장 메모리들이 존재한다. 이러한 메모리들을 효율적으로 테스트 하기 위해 BIST가 각 메모리에 적합한 마치 알고리즘을 제공해 주어야 한다. 즉, 다양한 테스트 알고리즘을 적용 가능한 programmable MBIST 구조가 효과적이다.

본 논문에서는 SoC안에 포함된 많은 수의 다양한 내장 메모리에 대해 서로 다른 마치 알고리즘으로 동시에 테스트 할 수 있는 새로운 programmable MBIST 구조와 그 방법론을 제안한다.

II. 본론

2.1 Memory BIST Architecture

SoC안의 다양한 메모리에 대해서 각각 별개의 테스트 알고리즘을 적용하여 보다 테스트 시간을 단축시키기 위해, 본 논문에서는 기존의 distributed MBIST 구조의 단점을 보완한 새로운 MBIST 구조를 제안하였다.

[2]에서 제안된 기존의 distributed MBIST 구조는 SoC안의 다양한 메모리에 대해 서로 다른 마치 알고리즘을 적용할 수 있지만, 다른 알고리즘을 적용할 경우 각 메모리들을 동시에 테스트 할 수 없다. 이 단점을 보완하기 위해서 본 논문에서 제안한 MBIST 구조는 IEEE Std. 1500 인터페이스를 사용해 마치 시퀀스 명령어를 각 메모리 BIST wrapper에 인가해 주는 방법으로 동시 테스트를 가능케 한다. 이 방법은 명령어를 각 wrapper에 인가해 주는 데에 약간의 시간이 필요하지만, 이는 전체 메모리 테스트 시간에 비하면 매우 적은 시간이기 때문에 무시할 수 있다. 또한 기존의 MBIST 구조는 테스트 중 고장이 발생했을 때 수행 중이던 테스트를 끝나치고 그 후에 고장진단을 수행하기 때문에 추가적인 고장진단 시간이 필요하다. 하지만 본 논문에서 제안하는 MBIST 구조는 고장 발생 시 이를 바로 BIST 컨트롤러에 알려주고 고장 정보를 ATE로 전달하기 때문에 추가적인 고장진단 시간을 절약할 수 있다.

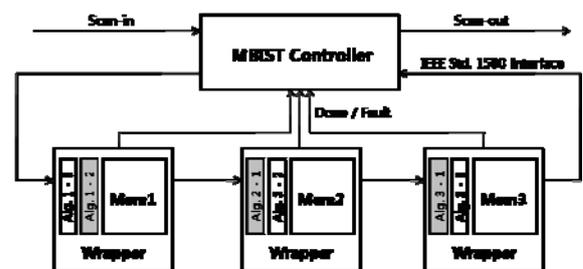


그림 1. 제안하는 MBIST 구조도

그림 1은 제안하는 MBIST 구조의 간략한 구조도이다. 그림에서 보듯이 MBIST 컨트롤러가 다양한 메모리 BIST wrapper에 대해 서로 다른 마치 시퀀스 명령어를 인가해 주어서 각 메모리가 서로 다른 마치 알고리즘으로 동시에 테스트 될 수 있다. 또한 BIST wrapper와 BIST 컨트롤러 사이에 직접적인 인터커넥션이 있어, 이를 통해 수행되고 있었던 마치 시퀀스의 완료 여부와 고장 여부를 BIST 컨트롤러가 알 수 있다. 그리고 고장이 발생하였을 경우, IEEE Std. 1500 인터페이스를 통해서 고장 정보 레지스터의 고장 정보를 컨

트롤러로 전달한다.

2.2 Memory BIST Controller

메모리 BIST 컨트롤러는 ATE로부터 마치 시퀀스 명령어를 받기 때문에 내부에 명령어를 저장할 별도의 대용량 메모리가 필요 없게 되어, 면적 오버헤드를 줄일 수 있다. MBIST 컨트롤러의 역할은 다음과 같다.

- 각 메모리에 대한 마치 시퀀스 명령어를 ATE로부터 받는다.
- IEEE Std. 1500 인터페이스를 통해 명령어를 각 wrapper로 전송한다.
- 마치 알고리즘이 수행 중 고장이 발생했을 때 고장 신호를 wrapper로부터 받으면, 해당 wrapper로부터 메모리 ID와 고장 정보를 받아 ATE에 전송한다.

2.3 Memory BIST Wrapper

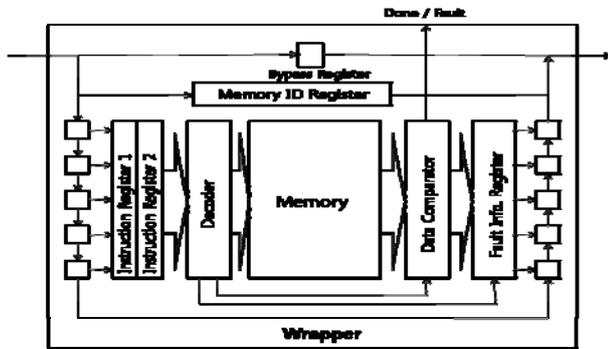


그림 2. 제안하는 MBIST Wrapper

그림 2 는 제안하는 MBIST wrapper의 구조도이다. wrapper는 두 개의 명령어 레지스터를 가지고 있다. 그 이유는 하나의 레지스터에 이미 명령어가 저장되어 있을 경우 wrapper로 들어온 새로운 명령어를 저장하기 위해서이다. 즉, 현재 수행되고 있는 마치 시퀀스가 완료 될 경우 바로 다음 시퀀스가 지연시간 없이 연속적으로 수행 될 수 있다. 디코더는 명령어 레지스터에 저장되어 있는 명령어를 이용해 마치 시퀀스를 위한 데이터 패턴과 주소 패턴을 생성한다. 데이터 비교기는 예상되는 결과와 실제로 메모리에서 나온 결과를 비교하여 그 결과를 컨트롤러로 전달한다. 또한 고장이 발생했을 경우 고장 신호를 컨트롤러로 전달하고 고장 정보 레지스터에 저장된 고장 정보도 메모리 ID와 함께 ATE로 전달한다.

MBIST wrapper에 의한 면적 오버헤드를 줄이기 위해 SoC안의 같은 종류의 메모리들에 대해서는 하나의 클러스터로 묶어 단일 MBIST wrapper를 썬어 준다.

III. 결과

위에서 언급한 distributed programmable MBIST 외에 단일 메모리에 대한 테스트를 수행하는 standalone programmable MBIST의 경우, 다양한 메모리를 서로 다른 알고리즘으로 테스트 하기 위해선 각 메모리를 순차적으로 테스트 해야 한다. standalone programmable MBIST, 기존의 distributed programmable MBIST [2]와 본 논문에서 제안한 MBIST를 적용하여 다양한 메모리들에 대해 서로 다른 알고리즘으로 테스트 할 때 그 성능을 정성적으로 비교한 결과를 표 1에 정리하였다. standalone MBIST의 경우 각 메모리들을 동시에 테스트 할 수 없으므로 테스트 시간이 오래 걸리고, 면적 오버헤드 또한 크다. 기존의 distributed MBIST의 경우 같은 테스트 알고리즘을 적용할 경우에는 메모리들을 동시에 테스트 할 수 있지만, 서로 다른 알고리즘을 적용할 시에는 동시 테스트가 불가능 하다. 본 논문에서 제안한 MBIST는 다양한 메모리에 대해 서로 다른 알고리즘을 동시에 적용하여 테스트 할 수 있으므로 테스트 시간이 가장 짧고, 면적 오버헤드 역시 낮다.

Criteria	Standalone MBIST	Distributed MBIST [2]	Proposed MBIST
Test time	Long	Medium	Short
Synchronous Test	No	Conditional	Yes
Area overhead	High	Low	Low

표 1. Existing MBIST vs. Proposed MBIST

IV. 결론 및 향후 연구 계획

본 논문에서 제안한 MBIST 구조는 다양한 메모리에 대해 동시에 서로 다른 테스트 알고리즘을 적용할 수 있다. 제안한 방법은 SoC의 내장된 메모리 테스트에 있어서 테스트 시간을 감소시키면서 테스트 효율성을 향상 시킬 수 있다.

앞으로는 본 논문에서 제안한 구조를 RTL을 통해 실제 하드웨어로 구현하여 검증하는 연구를 진행하고자 한다.

참고문헌

- [1] D. Appello, V. Tancorre, "Embedded Memory Diagnosis: An Industrial Workflow", IEEE International Test Conference, pp.1-9, 2006.
- [2] M. L. Bodoni, A. Benso, S. Chiusano, S. D. Carlo, G. D. Natale, and P. Prinetto, "An Effective Distributed BIST Architecture for RAMS", IEEE European Test Workshop, pp. 119-124, 2000.