논문 2004-41SD-9-12

SOC 테스트 시간 축소를 위한 새로운 내장 코어 기반 SOC 테스트 전략

(A New Test Technique of SOC Test Based on Embedded Cores for Reducing SOC Test Time)

강 길 영*, 김 근 배**, 임 정 빈**, 전 성 훈**, 강 성 호**

(Gil-Young Kang, Gun-Bae Kim, Jung-Bin Im, Sung-Hoon Chun, and Sung-Ho Kang)

요 약

본 논문에서는 내장 코어 기반 SOC의 테스트를 위한 새로운 테스트 전략을 제안한다. SOC 테스트는 전체 테스트 시간을 얼마나 줄일 수 있는가에 따라서 그 성능을 평가할 수 있다. SOC를 구성하는 코어에 대한 테스트 시간은 코어에 구성된 테스트 래퍼 구조에 의해서 결정되며, 테스트 래퍼는 TAM을 사용하기 때문에 결국 TAM에 할당되어 있는 스캔 체인의 길이에 의해서 결정된다. 따라서 SOC 설계 단계에서 테스트를 고려한 설계가 이뤄져야 하며 효율적인 테스트를 위해서는 테스트 전략을 잘 세워야 한다. 기존의 테스트 기법은 모두 SOC 전체 TAM 라인들을 몇 개의 그룹으로 나누고 코어에 할당된 스캔 체인들을 TAM 라인에 적절히 분배해서 코어의 테스트 시간과 SOC 전체의 테스트 시간을 모두 최소화 할 수 있는 구조를 만드는 방법이었다. 하지만 이는 NP 문제로 모든 조합에 대한 시도를 통해서 최적의 결과를 찾는 것이 불가능하다. 본 논문에서는 이 문제에 대한 새로운 방법을 제안하고 그 효율성을 증명한다.

Abstract

A new test strategy for embedded SOC test is proposed. The SOC test is evaluated by the degree that is the amount of the total reduced test time. Since the test time for a embedded core is determined by the configuration of test wrapper, the total test time is decided by the length of the largest TAM used by the test wrapper. So the DFT(Design for Test) must be involved in the design flow. And the efficient test strategy must be settled. The all previous test strategies are the methods that find a sub-optimal configurations of scan-chains to minimize the test time after the total TAM lines are divided into a few groups. But this is the NP-complete problem so that all attempts which examine such a TAM configuration and scan-chain division are impossible. In this thesis, a new methodology for this problem is proposed and the efficiency of the methodology is proved.

Keywords : SOC 테스트, TAM, 스케줄링

I. 서 론

SOC 형태의 설계 방식이 새로운 설계 형식의 표준으로 자리를 잡아가면서 관련된 많은 분야에 대해서 연구가 이뤄지고 있다. 그 중에서 SOC 테스트가 가장 활발한 연구 활동을 보이고 있다. SOC 테스트는 칩의 테스트 시간 단축, 테스트를 위해서 필요한 하드웨어 오버헤드의 최소화 및 소모 전력을 줄이기 위한 방법들에 대한 연구들이 주를 이루고 있다. 특히 SOC 테스트 시간을 줄이기 위해서 많은 연구들이 이뤄지고 있는데, 대표적으로 SOC 테스트 스케줄링 분야가 있다. SOC 테스트 스케줄링은 쉽게 말해서 SOC의 구체적인 테스트 방법과 순서들에 대한 연구를 말하는데, 일반적으로 P1500[1]을 기반으로 하는 상황에서 TAM 스케줄링과

* 정회원, 삼성전자 반도체총괄 메모리사업부
(Samsung Electronics, Co. Ltd)
** 정회원, 연세대학교 전기전자공학과
(Department of Electronics Engineering, Graduate School, Yonsei University)
※ 본 연구는 한국과학재단 목적기초연구 (과제번호: R01-2003-000-10150-0) 지원으로 수행되었음.
접수일자: 2003년8월12일, 수정완료일: 2004년9월2일

같은 의미로 사용되고 있다.

TAM 스케줄링은 SOC를 구성하는 각각의 IP 형태의 내장되는 코어들에 대한 테스트 전략을 세우는 과정에서 가장 중요한 부분을 차지한다. 이 전략에 따라서 SOC 전체의 구조가 변할 수도 있고, 제공되는 테스트 패턴 세트를 수정하기도 한다. 또한 전체 테스트 제어를 위해서 필요한 추가적인 테스트 비용이 발생할 수도 있다. 그래서 보통 TAM 스케줄링에서는 P1500 표준을 사용해서 이러한 문제들을 해결한다. 표준안을 사용하면 전체적인 테스트 전략에 대한 고민 없이 TAM 설계만으로 SOC 전체에 대한 테스트 방법을 제시할 수 있기 때문이다.

TAM 스케줄링에서 가장 기본이 되는 것은 TAM이 할당된 코어에 대해서 테스트 시간을 최소한으로 만드는 것이다. 테스트 시간은 테스트 래퍼의 구조에 따라서 결정되며, 그 이유는 하나의 코어에 대한 테스트를 위해서 코어의 입출력 터미널의 수와 내부 스캔 체인의 개수에 따라 테스트 래퍼의 구조가 바뀌기 때문에 코어에 대한 테스트 시간 또한 전적으로 테스트 래퍼의 구조에 따라서 결정된다. 예를 들어 10개의 스캔 체인으로 구성되는 코어에 대해서 TAM 라인이 5개 할당되는 경우와 10개가 할당되는 경우에는 내부의 스캔 체인에 테스트 벡터를 채우는 시간과 순서가 서로 다르다. 전자의 경우에는 하나의 TAM 라인을 이용해서 두 개의 스캔 체인을 제어해야 하며, 후자의 경우에는 각 스캔 체인들을 독립적으로 제어할 수 있다. 따라서 테스트 시간 또한 차이를 보이게 된다. 일반적으로 TAM 라인의 수가 클수록 테스트 시간은 더 작아지지만 핀 오버헤드가 커지는 상충관계를 갖는다.

코어에 할당되는 TAM의 크기가 결정되면, 코어의 테스트 시간을 최소화하기 위해서 구성되는 스캔 체인의 길이가 평형 상태(balancing)를 유지해야 한다. 평형 상태란 스캔 체인들의 길이가 모두 일정한 상태로 여기서는 TAM에 할당된 TAM 라인의 길이가 모두 같은 경우를 말한다. 따라서 테스트 시간은 코어내의 최대 스캔 체인의 길이에 의해 결정되며[2], 평형 상태일 경우에 테스트 시간이 최소가 된다. 만약 SOC 설계자에게 제공되는 코어가 소프트 코어의 형태라면 코어의 스캔 체인을 평형 상태로 유지하는 것이 가능하다. 하지만 대부분의 IP 공급자들은 하드 코어의 형태로 IP를 제공하는 것이 일반적이며[3], 스캔 체인들의 길이 또한 결정된 상태로 제공된다. 따라서 SOC 설계자는 스캔 체인을 재구성하는 방법을 필요로 하게 된다.

스캔 체인 재구성의 목적은 주어진 스캔 체인들을 가지고 할당된 TAM 라인에 적절히 분배해서 코어의 스캔 체인들의 길이를 평형 상태로 만드는 것이다. 이것은 NP-complete 문제이며[4], 체인의 수가 많아질수록 최적의 조합을 찾는 것이 힘들어진다. 가장 간단한 방법으로는 할당된 TAM에 스캔 체인을 순서대로 분배하면서 나누는 방법[4]이 있다. 이 방법은 비교적 평형 상태와 비슷한 결과를 얻으면서 빨리 조합을 얻어낼 수 있다는 장점이 있다. 또 가장 길이가 작은 TAM 라인에 스캔 체인을 할당해 나가는 방식[5]도 있는데, 이 방법은 앞의 방식보다 좀 더 좋은 결과를 얻을 수 있다. 하지만 NP 문제들이 최적의 해답을 얻기 위해서 많은 시간이 필요하며 얻어낸 결과가 최적의 해답인지 여부를 알 수 없기 때문에, 위의 두 방식은 최적의 조합을 찾아내지 못하는 경우가 있다. 따라서 많은 새로운 연구들이 진행 중에 있으며, Bin-Packing 알고리즘[6]이나 Simulating Annealing 알고리즘[7]을 이용하여 최적의 조합을 찾으려는 연구들이 발표되고 있다.

본 논문에서는 최적의 스캔 체인 구성을 위해서 스캔 체인의 평형 상태를 찾을 수 있는 방법을 제안하고 실제로 스캔 체인의 구성 방법에 따라서 SOC 테스트 시간의 차이가 아주 크게 나타남을 보여준다. 이를 위해 다음과 같이 구성되는데, Ⅱ장에서는 기존의 알고리즘들을 소개하고, Ⅲ장에서는 새롭게 제안하는 SOC 테스트 방법을 소개한다. 그리고 이를 적용했을 때, 얻을 수 있는 결과를 Ⅳ장에서 정리하고 있다. 그리고 Ⅴ장에서 본 논문의 결론을 마무리 짓는다.

Ⅱ. 기존 연구

스캔 체인의 길이는 테스트 시간과 직접적으로 비례한다. 따라서 테스트 시간을 줄이기 위해서는 스캔 체인의 길이를 최소한으로 줄이는 것이 중요하다. 일반적으로 스캔 체인의 길이는 TAM의 크기가 커질수록 작아진다. 하지만 전체 스캔 체인을 구성하는 각 세부 체인들의 크기에 따라서 이러한 현상은 변할 수 있다. 즉 전적으로 TAM의 크기에 따라서 해당 코어의 스캔 체인의 길이가 결정되며, 가장 최적의 조합을 찾는 것은 NP 문제로 분류된다[4].

SOC를 설계하는 IP 사용자에게 제공되는 IP 형태의 코어들은 소프트 IP 형태와 하드 IP 형태, 두 가지로 나눠진다. 소프트 IP의 경우에는 [2]와 같이 사용자의 의도대로 최적의 테스트 방안이 결정된다. 하지만 IP는

일반적으로 하드 IP 형태이기 때문에, SOC 테스트 시간을 최소화하기 위해서는 사용할 수 있는 TAM 라인을 가능한 최대로 사용하면서 각 코어를 구성하는 스캔 체인의 최대 길이를 최소로 구성해야 한다.

하나의 코어에 대해서 내부의 기존 스캔 체인들을 묶어서 다수의 스캔 체인을 구성하는 경우 필요한 정보는 코어 내의 스캔 체인의 개수와 각 체인들의 길이, 재구성할 스캔 체인의 개수이다. 재구성할 체인의 개수는 TAM 라인 수와 일치한다고 가정한다. 즉 사용자가 사용할 수 있는 TAM 라인을 최대한 이용해서 구성되는 스캔 체인들 중에서 최대 길이를 최소로 만드는 것이 SOC의 테스트 시간을 줄이기 위한 가장 간단한 방법이다. 이러한 방법들은 여러 가지 형태로 연구되어 왔으며 다음과 같은 방법들이 알려져 있다.

1. LPT (Largest Processing Time) 알고리즘

SOC 테스트의 대표적인 알고리즘은 Graham이 제안한 LPT (Largest Processing Time) 알고리즘[8]이다. LPT 알고리즘을 이용한 스캔 체인 구성 방법은 다음과 같다. 일단 해당 코어 내부의 스캔 체인들의 길이를 기준으로 내림차순으로 정리한다. 그리고 각각의 스캔 체인들을 현 상태에서 체인들의 길이의 합이 가장 작은 TAM 라인에 할당하는 방식으로 배당한다. 예를 들어 SOC 벤치 회로인 P34392의 10번 코어[9]에 대해서 TAM = 3인 경우에 재구성되는 스캔 체인 구조를 결정하기 위해서 LPT 알고리즘을 적용하면, {(519, 429, 362, 268), (501, 438, 393, 64, 54, 36, 36, 28, 20), (501, 480, 286, 276, 24, 16)}으로 결정되고, 최대 스캔 체인의 길이는 3번 TAM 라인의 길이인 1583이 된다. 그림 2-1은 LPT 알고리즘의 의사 코드이다.

2. FFD (First Fit Decreasing) 알고리즘

FFD (First Fit Decreasing) 알고리즘[4]은 LPT 알고리즘과 비교했을 때, 그 사용 목적이 약간 다르다. FFD 알고리즘을 이용해서 스캔 체인을 구성하는 경우는 스캔 체인의 최대 길이가 정해지는 경우에 사용된다. 그리고 전체 체인을 구성하기 위해서 필요한 TAM 라인의 최소 크기를 결정한다. FFD 방식은 TAM이 허용할 수 있는 최대 스캔 체인의 길이가 정해지면, LPT 방식에서와 마찬가지로 길이의 내림차순으로 정렬된 스캔 체인들을 허용 한도 내에서 연속적으로 할당한다. 만약 체인들을 첫 번째 TAM 라인의 허용 한도 내에서 최대로 채웠다면, 남아있는 스캔 체인들을 두 번째 라인에 채워 넣으며, TAM 대역폭 내에 모두 할당할 수 있으면 알고리즘은 종료된다. FFD를 응용한 한도 내에서 연속적으로 할당해나가면서 부적합한 체인이 나타나면 다음 순번의 체인을 할당하는 방식은 더 효율적인 FFD 알고리즘을 지원한다.

LPT 알고리즘의 예에서 설명한 P34392의 10번 코어에 대해서 FFD 방식을 적용하면, 최대 스캔 체인의 길이를 1580으로 정한 후 TAM의 대역을 정하지 않는 경우에 대해서 {(519, 501, 501, 54), (480, 438, 429, 64), (393, 362, 286, 276, 36, 36, 28, 24, 20, 16), (268)}을 찾아낸다. 여기서 최대 한계뿐만 아니라 최소 한계까지 정해주면 위의 결과보다 더 효율적인 결과를 찾을 수 있지만 실행 시간은 지수적으로 커지며, TAM의 대역폭이 정해지면 알고리즘은 비효율적인 실행을 하게 된다. 그래서 일반적으로 FFD 알고리즘은 Pass/Fail을 판별하는 용도로 사용하며, 실제 스캔 체인 구성을 위한 스케줄링은 LPT방식을 응용한다. 그림 2-2는 FFD 알고리즘의 의사 코드이다.

3. MULTIFIT 방법

위에서 설명한 LPT 방식과 FFD 방식이 실제로 어떻게 체인들을 구성할지 결정하고, 설계자의 목표에 맞는 디자인을 찾기 위한 알고리즘이라면, MULTIFIT 알고리즘은 TAM 라인의 한계 길이를 결정하는데 응용

```
(for S={S1, S2 ..., Sy; y=#of scan-chains}, m=#of TAM lines)
sort S such that length(S1)≥length(S2)≥ length(S3)≥... ≥length(Sy);
for i = 1 to m
  assign TAMi = Si;
for i = (m+1) to y
  find the minimum TAM line, TAMj;
  assign Si into TAMj;
return Max_TAM_length and # of TAM_line;
```

그림 2-1. LPT 알고리즘의 의사 코드 (Pseudo Code)
Fig. 2-1. Pseudo Code for LPT Algorithm.

```
(for S={S1, S2 ..., Sy; y=#of scan-chains}, m=#of TAM lines)
sort S such that length(S1)≥length(S2)≥ length(S3)≥... ≥length(Sy);
for i = 1 to m
  assign TAMi = 0;
for i = 1 to y
  assign j = 1;
  while (length(TAMj) + length(Si) > C)
    assign j = j + 1;
  assign Si into TAMj;
return MAX{j | length(TAMj) ≠ 0}
```

그림 2-2. FFD 알고리즘의 의사 코드 (Pseudo Code)
Fig. 2-2. Pseudo Code for FFD Algorithm.

될 수 있다. MULTIFIT 알고리즘은 FFD 알고리즘을 응용한 것으로 이미 알려진 상한 한계와 하한 한계를 초기 값으로 설정하고, 각 설정마다 FFD를 이용해서 최상의 스캔 체인들을 할당할 수 있는지 판별하고 이 작업이 실패할 때까지 새로운 한계를 찾는 알고리즘이다[10]. 일반적으로 하한 한계는 전체 플립플롭의 수의 평균으로 구하고 상한 한계는 하한 한계의 2배 크기를 설정한다[4]. MULTIFIT 알고리즘은 LPT 알고리즘에 비해서 비교적 시간이 오래 걸리는 단점이 있으나 스캔 체인 구성에서 생길 수 있는 최악의 경우를 방지할 수 있다. 하지만 MULTIFIT 방식은 모든 경우에 대해서 좋은 조합만을 찾아주는 것이 아니기 때문에 비교적 빠른 시간 내에 적당한 조합을 찾는 경우에 사용될 수 있다.

4. COMBINE 알고리즘

COMBINE 기법은 스캔 체인들에 대해서 LPT 알고리즘의 결과를 구하고 조건에 따라서 위에서 설명한 LPT 방식과 FFD 방식 중에서 하나를 선택하여 사용하는 방법이다[4]. LPT 방식이 스캔 체인 구성에서 한계값을 찾는데 유효하고, FFD 방식이 정해진 한계에서 TAM 라인의 사용을 줄일 수 있다는 장점을 이용해서 두 방식을 선택적으로 사용한다. 두 방식을 선택하는 기준은 여러 가지 방법이 있을 수 있지만 기존에 발표된 방식은 MULTIFIT 방식에서 설명한 하한 값을 기준으로 LPT 결과가 하한의 1.5배 이상이면 LPT 결과를 사용하고 미만일 경우에는 FFD를 반복적으로 실시하여 최적의 값을 찾는 방법이다.

Ⅲ. 새롭게 제안하는 SOC 테스트 시간 단축을 위한 테스트 전략

Ⅱ장에서는 다양한 스캔 체인 구성 방식에 대해서 알아보았다. 기존의 방법들은 NP 문제인 체인 구성 문제를 선형 해석 가능한 형태로 모델링하고, 기존의 잘 알려진 상자 채우기 알고리즘을 이용하여 답을 찾는 방식이다. 본 논문에서는 래퍼 구성 과정에서 필수적인 스캔 체인의 길이 최소화를 위해서 기존의 LPT 알고리즘을 응용해서 사용한다. 이를 MLPT라고 정의하고, MLPT를 통해서 성능 향상이 어려운 경우를 위해서 TAM 분할 방법을 고안한다. 또 실제 테스트 상에서 필요한 TAM 라인의 수를 줄이는 방법으로 최적의 래퍼 구조를 지원하는 스캔 체인 구성 방법을 제안한다.

1. (Step 1) Reconfigurable Scan Chain Optimization

LPT 방식은 TAM 라인에 할당되는 스캔 체인들의 길이를 가능한 비슷하게 맞추기 위해서 가장 길이가 짧은 TAM 라인에 스캔 체인들을 할당하는 방법이다. 예를 들어 6개의 스캔 체인을 가지는 코어가 있다고 가정하고 각각의 길이가 {12, 5, 17, 9, 14, 4}라고 가정한다. 이럴 경우 TAM의 대역폭은 최소 1에서 최대 6까지 가능하며, 각각의 경우 LPT 기법을 이용해서 스캔 체인을 나누는 방법은 다음과 같다.

TAM Bandwidth	Scan Chain Partition	Maximum Scan Chain Length
1	{12, 5, 17, 9, 14, 4}	61
2	{12, 5, 14} {17, 9, 4}	31
3	{12, 9}, {5, 14}, {17, 4}	21
4	{17}, {12, 4} {5, 9}, {14}	17
5	{17}, {14} {9}, {12}, {5, 4}	17
6	{17}, {14}, {9} {12}, {5}, {4}	17

스캔 체인을 나누는데 있어서 가장 중요한 것은 최대 스캔 체인의 길이를 최소한으로 줄여야 한다는 것이다. Ⅱ-1에서 LPT 알고리즘은 코어를 구성하는 체인들의 특성과 TAM 라인의 수에 관계없이 비교적 최적화된 결과를 구할 수 있음을 설명했다.

하지만 LPT 방식은 순차적으로 체인을 할당하기 때문에 그 결과가 최적의 값은 아니다. 예를 들어 Ⅱ-1에서 설명한 P34392의 10번 코어에 대해서 TAM = 3인 경우에 LPT 결과를 보면 3번 TAM 라인의 길이인 1583이 최대 길이가 된다. 하지만 2번 라인의 길이가 24인 스캔 체인과 3번 라인의 길이가 20인 체인을 서로 바꿔주면 전체 최대 스캔 체인의 길이는 3번 라인의 1579가 된다. 모든 체인의 길이를 더해서 TAM 라인의 수로 나눈 평균이 1577임을 고려할 때, 테스트 시간이 더 줄어드는 정도를 알 수 있다.

본 논문에서는 LPT 알고리즘의 결과를 초기 값으로 설정하고 다음과 같은 방식으로 결과를 조정해 나간다. 우선 LPT의 결과를 분석한다. 만약 스캔 체인 할당이 적절히 이루어진다면, TAM 라인에 할당된 스캔 체인

```
(for S={S1, S2 ..., Sy; y=#of scan-chains}, m=#of TAM lines)
do LPT;
while (Max_TAM_length_new ≠ Max_TAM_length_old)
  assign k = Max_TAM_line;
  for i = 1 to m (except i = k)
   for all scan-chains in i^th TAM line
    if (Max_TAM_length_new < Max_TAM_length_old)
      find Max_difference(Sα, Sβ); // Sα = element in k^th TAM line
                                   // Sβ = element in i^th TAM line
  exchange Sα and Sβ;
```

그림 3-1. MLPT 알고리즘의 의사 코드 (Pseudo Code)
Fig. 3-1. Pseudo Code for MLPT Algorithm.

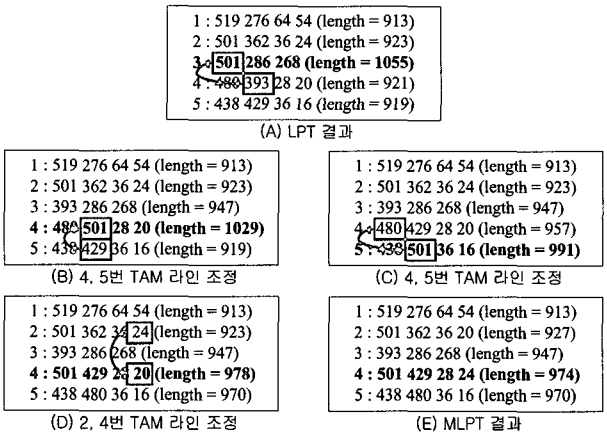


그림 3-2. MLPT 적용 예제
Fig. 3-2. The exercise of MLPT application.

의 최대 길이는 평균값과 가깝게 된다. 즉 평균값과 최대 체인의 길이 차가 작을수록 더 좋은 조합이 되며, 이를 위해서 길이가 가장 긴 TAM 라인을 구성하는 스캔 체인들과 바꿀 수 있는 체인을 찾는다. 대상 체인을 찾는 경우에는 하나씩 바꿀 수 있으며 여러 개를 묶어서 바꿀 수도 있다. 이 방법을 MLPT (Modified Largest Processing Time) 알고리즘이라고 정의한다. 그림 3-1은 MLPT 알고리즘을 의사 코드로 나타낸 것이다.

그림 3-2는 P34392의 10번 코어에 대해서 TAM 라인이 5인 경우에 제안하는 MLPT를 적용한 예이다. (A)를 보면 LPT 알고리즘의 결과로 체인의 TAM 라인의 평균 길이인 946을 중심으로 최소 913에서 최대 길이인 1055의 구성을 가지는 조합을 얻을 수 있다. 여기서 가장 길이가 긴 3번 라인의 1번 스캔 체인을 4번 라인의 2번 체인과 바꿔서 구성하면 (B)와 같이 최대 TAM 라인의 길이가 1029인 구성을 얻을 수 있게 된다. 다시 4번 라인의 이런 식으로 (C), (D)의 과정을 거쳐서 반복하면, (E)와 같이 최대 체인의 길이가 974인 조합을 찾을 수 있다. 이 경우 체인의 길이가 1055에서 974로 81만큼 약 7.7% 줄어든다. 그리고 이 코어의 테스트를 위해서 454개의 테스트 패턴이 필요하기 때문에 테스트 시간은 [2]의 방법을 사용하는 경우 480479 bits에서 443624bits로 역시 약 7.7% 줄어든다. [2]의 특징은 테스트 패턴이 많은 코어의 경우 체인의 길이를 줄여서 얻을 수 있는 이득이 더 커진다는 점이다.

표 3-1. 평형 상태의 스캔 체인에서 LPT와 MLPT의 결과
Table 3-1. LPT and MLPT result for balanced Scan-Chain.

TAM Bandwidth	1	2	3	4	5	6	7	8	9	10
LPT	1122	570	441	331	240	221	220	220	220	130
MLPT	1122	570	440	330	240	221	220	220	220	130

표 3-2. 체인의 수가 작은 경우의 LPT와 MPLT의 결과
Table 3-2. LPT and MLPT result for small Scan-Chain.

TAM Bandwidth	1	2	3	4
LPT	209	108	99	99
MLPT	209	108	99	99

2. (Step 2) TAM Assignment

MLPT 방법이 항상 LPT 방식보다 더 좋은 결과를 얻을 수 있는 것은 아니다. MLPT는 최대 길이를 구성하는 스캔 체인들 중에서 두 개의 후보를 선정해서 바꾸는 과정을 반복하는데, 만약 코어를 구성하는 체인들이 평형 상태를 유지한다면, MLPT 방식뿐만 아니라 그 어떤 알고리즘으로도 LPT 이상의 결과를 얻을 수 없다. 또한 코어를 구성하는 체인의 수가 아주 작으면 MLPT 방식을 적용해도 더 좋은 조합을 찾지 못한다. 표 3-1은 P22810의 1번 코어[9]에 대해서 LPT 방식과 MLPT 방식을 적용했을 경우 최대 TAM 라인의 길이를 나타낸 것이다. 이 코어는 총 10개의 스캔 체인으로 구성되어 있으며, 각 체인의 길이는 {130, 111, 111, 110, 110, 110, 110, 110, 110, 110}로 체인들의 길이가 비교적 일정하다. 결과를 보면 알 수 있듯이 코어를 구성하는 체인의 길이가 처음부터 평형 상태를 유지하면, 최적의 스캔 체인 구조는 LPT 알고리즘의 결과임을 알 수 있다.

또 다른 예로 P22810의 10번 코어는 {99, 98, 10, 2}의 스캔 체인을 가지며, 이에 대한 두 가지 방식에 의한 결과가 다음과 같다. 즉 체인의 수가 작은 경우에도 MLPT의 효과를 기대하기는 어렵다는 것을 알 수 있다. 표 3-2가 이런 점을 잘 보여 준다.

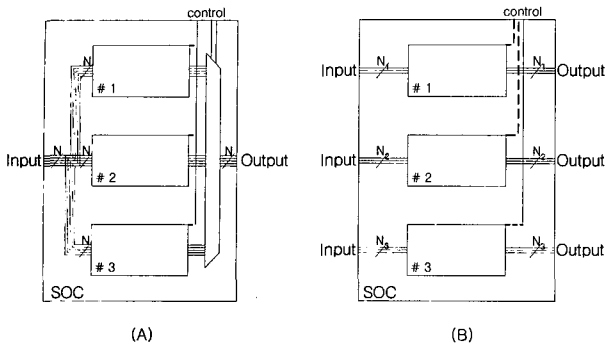


그림 3-3. 다중 구조와 분배 구조
Fig. 3-3. Multiplex and Distributed Architecture.

이와 같이 Ⅲ-1에서 제안하는 MLPT 알고리즘을 통해서 모든 경우에 대하여 테스트 시간과 직접적으로 연관 있는 최대 TAM 라인의 길이를 줄이지는 못한다. 따라서 다수의 코어들로 이뤄지는 SOC 테스트를 하는 경우 체인의 길이를 조정하는 것이 아니라 TAM을 테스트 대상 코어들에 대해서 적절하게 할당해서 테스트 시간을 조정하는 스케줄링이 필요하다. 테스트 스케줄링에서 결정하는 것은 테스트 대상 코어들의 배치 및 순서, 그리고 TAM 분배 방법 등이 있다. 여기서 가장 중요한 것이 칩이 사용할 수 있는 TAM 라인을 각각의 코어들에 적절하게 분배하는 방법이며, 그 이유는 테스트 순서와 코어의 배치가 결정되기 때문이다.

일반적으로 가장 쉽게 생각할 수 있는 TAM 할당 방식은 모든 테스트 대상 코어에 사용 가능한 TAM 라인을 모두 할당하는 방식이 있다. 이런 구조를 다중(Multiplexing) 구조라 하며, 테스트 패턴은 모든 코어에 동시에 가해지고, 이에 대한 응답을 다중화기(MUX)를 통해서 선별하는 동작을 한다. 이런 구조의 문제점은 모든 코어가 항상 테스트 모드에서 동작하기 때문에 소모 전력이 SOC 한계 전력을 넘어갈 수 있다. 또 모든 코어들에 대해서 TAM 라인이 연결되어야 하기 때문에 와이어링 오버헤드 문제가 생길 수 있다.

또 다른 방식으로 분배 구조가 있다. 분배 구조는 모든 테스트 대상 코어에 대해서 독립적으로 TAM을 할당하는 방식이다. 이 방식은 모든 대상 코어들이 독립적으로 동시에 테스트가 진행되며, 가장 시간이 오래 걸리는 코어에 의해서 테스트 시간이 결정된다. 이 방식은 다중 구조에서 생기던 소모 전력 문제가 없는 반면에 코어들 모두에 대해서 동시에 테스트 입력 및 응답 결과 확인이 이뤄져야 하기 때문에 많은 제어 블록들이 필요하다. 그림 3-3은 위에 설명한 다중 구조와 분배 구조를 간략하게 나타낸 것이다.

대부분의 SOC 테스트 스케줄링 방법들은 위에서 설명한 두 가지 방법들을 적절히 조합하는 형태로 생각할 수 있다. 즉 일부 테스트 대상 코어들을 하나로 묶어서 분배 구조 형태로 구성하고 각각의 그룹 내에서는 다중 구조 방식을 사용[11]하는 것이다. 여기서 중요한 것이 전체 TAM을 몇 개의 그룹으로 나누고 그 크기를 어떻게 정하는가에 따라서 코어들의 배치가 결정된다. 본 논문에서는 다음과 같은 방식으로 결정된다.

우선 TAM의 크기를 M이라 하고 코어의 개수를 C라고 한다. TAM의 크기 M이 C보다 작은 경우에는 TAM을 어떻게 분할한다고 해도 하나의 TAM 라인에 두 개 이상의 코어가 할당된다. 따라서 TAM은 최소 1개에서 최대 M개의 독립적인 TAM 그룹으로 나눠질 수 있다. 그러면 나눠진 TAM을 크기 순서로 정렬하고 여기에 코어들을 할당하는 방법을 사용할 수 있다. 기본적으로 상자 채우기 알고리즘을 응용할 수 있으며, 방식은 LPT 알고리즘과 동일하다. 즉 현재 가장 테스트 시간이 짧은 TAM 라인 그룹을 찾고 이것을 이용해서 테스트를 수행할 때 가장 시간이 오래 걸리는 코어를 선정해서 해당 TAM 라인 그룹에 할당하는 방식을 사용한다.

M이 C보다 큰 경우는 두 가지 방법이 있다. 첫 번째는 모든 코어들에 최소한의 TAM 라인을 할당하고 가장 테스트 시간이 오래 걸리는 그룹에 남은 TAM 라인을 할당해 나가는 방법이다. 이것은 LPT 알고리즘을 응용한 것으로 비교적 그 간단하다. 하지만 TAM 라인이 증가해도 코어의 최대 스캔 체인 그룹의 길이가 반드시 줄어드는 것은 아니다. 표 3-1에서 볼 수 있듯이 할당되는 TAM의 크기가 7이면 TAM 라인이 8이나 9로 증가해도 효과를 볼 수 없다. 따라서 이런 특성은 첫 번째 방법의 종결 조건으로 사용될 수 있다. 두 번째 방법은 미리 TAM을 할당하고 여기에 코어들을 할당해 나가는 방식이다. 이 방법은 M이 C보다 작은 경우와 동일한데, 문제는 TAM의 크기가 커질수록 알고리즘의 수행 시간이 지수적으로 증가한다는 단점이 있다. 본 논문에서는 스캔 체인의 구성 최적화를 통한 테스트 시간 단축을 위해서 전자와 후자의 방법을 동시에 사용한다. 전자의 방법이 비교적 빨리 결과를 얻을 수 있기 때문에 초기 값으로 설정하고 후자의 방법을 더 좋은 조합을 찾기 위한 방법으로 사용한다.

3. (Step 3) Reducing TAM Usage

Ⅲ-2에서 설명한 방법으로 TAM 분배가 이뤄지면

표 3-3. TAM=78에서 P22810의 TAM 할당 결과
Table 3-3. Example for TAM Distribution for TAM=78.

core ID	1	2	3	4	5	6	7	8	9	10	11
TAM assignment	10/10	5/29	4/24	1/4	1/8	1/11	1/4	1/3	1/6	1/1	1/4
Test Time(Bits)	**102965**	91958	101551	8189	56049	67209	561	8610	16073	989	3093
core ID	12	13	14	15	16	17	18	19	20	21	22
TAM assignment	3/5	1/3	1/4	6/10	1/3	1/7	1/5	6/18	29/31	1/1	1/5
Test Time(Bits)	80624	5958	28999	92267	9779	11889	5067	85967	101555	104	11285

대부분의 경우 다중 구조, 분배 구조는 물론이고 LPT 방식의 TAM 분배 방법보다도 더 성능 향상이 일어남을 Ⅳ에서 증명한다. 하지만 TAM의 크기가 전체 IP들의 스캔 체인의 수의 합에 가까워질수록 그 성능이 떨어진다. 그 이유는 모든 체인들에 TAM 라인이 하나씩 할당되면 굳이 TAM 분배를 할 필요가 없어지기 때문이다. 이럴 경우에는 TAM을 분배하고 여기에 테스트 대상 코어들을 할당하는 방법보다 사용할 수 있는 모든 최대 TAM 라인을 사용할 수 있다고 가정하고 단순하게 테스트 시간의 상자 채우기 알고리즘을 적용하는 것이 더 효율적이다. 다음의 예제를 살펴본다.

표 3-3은 P22810에 대해서 TAM = 78인 경우에 Ⅲ-2에서 설명한 방법으로 TAM을 할당하고 테스트 시간을 구한 것이다. 표 3-3에서 보면 알 수 있듯이 최대 테스트 시간은 1번 코어에 의해서 결정되며, 더 이상 테스트 시간을 줄일 수 없다. 즉 TAM의 크기가 커지면 특정 코어에 의해서 테스트 시간이 결정되는데, 이런 경우에는 적절하게 코어들을 같은 테스트 그룹으로 묶음으로써 TAM 사용양을 줄일 수 있다. 예를 들어 7번 코어는 크기가 1인 TAM을 사용하고 있다. 만약 7번 코어가 8번 코어와 같은 테스트 그룹으로 엮여 있다면, TAM 사용하는 TAM 라인을 줄일 수 있다.

같은 TAM을 사용해서 테스트 되는 코어들의 그룹을 정하는 방법으로 FFD 알고리즘을 사용한다. 우선 테스트 시간의 크기를 기준으로 정렬한 후 최대 테스트 시간을 넘지 않으면서 같은 TAM을 사용하는 코어들을 같은 TAM에 할당해 준다. 만약 기존에 TAM을 사용하지 못하는 경우라면 테스트 그룹을 만들 수는 없지만 할당된 TAM의 크기가 비교적 작은 경우에는 가능하다. 표 3-4는 표 3-3에 대해서 FFD를 적용하고 같은 TAM을 이용해서 테스트되는 코어들을 정리한 것이다.

표 3-4. TAM 사용 축소 예
Table 3-4. Reduction of TAM Usage.

Test Group	1	2	3	4					5		
core ID	1	2	3	4	7	10	19	21	5	9	17
TAM Assignment	**10/10**	**5/29**	**4/24**	4/4	4/4	1/1	**6/18**	1/1	**1/1**	1/1	1/1
Core Test Time(Bits)	**102965**	91958	101551	3899	209	989	85967	104	56049	16073	11889
Group Test Time(Bits)	**102965**	91958	101551	91168					100363		

Test Group	5		6			7				8	9
core ID	22	18	6	14	13	8	11	12	16	15	20
TAM Assignment	1/1	1/1	**1/1**	1/1	1/1	3/3	3/4	**3/5**	3/3	**6/10**	**29/31**
Test Time(Bits)	11285	5067	67209	28999	5958	8065	2339	80624	4679	92267	101555
Group Test Time(Bits)			102166			95707				92267	101555

78개의 TAM 라인을 사용하는 경우와 테스트 시간은 동일하지만 TAM 사용 양을 65로 줄일 수 있다.

표 3-4를 보면 TAM Assignment에서 볼드체로 나타낸 것이 각 그룹을 구성하는 시드 코어가 된다. 즉 시드 코어에 할당된 TAM 라인의 수보다 적은 TAM 라인을 사용하는 코어들을 하나의 그룹으로 구성한다. 그리고 구성의 조건은 이미 결정된 1번 코어의 테스트 시간을 초과하지 않는 범위에서 결정되어야 한다.

Ⅳ. 실험 결과

Ⅲ에서 제안한 방법을 ITC'02 SOC 테스트 벤치 회로[9]에 적용했다. 그 결과가 표 4-1이다. 실험 대상 벤치 회로들은 모두 다수의 스캔 체인 기반 코어들로 구성되며, 각각에 대한 테스트 벡터가 제공된다. 그리고 기존에 발표된 SOC 테스트 방법을 적용한 결과와 비교해 보았다. 결과를 보면 대부분의 경우 TAM의 최대 길이를 줄임으로써 기존 알고리즘 대비 테스트 시간이 전체적으로 10~40% 감소함을 알 수 있다.

회로별 특징을 살펴보면, P34392의 경우에는, 스캔 체인으로 구성된 코어의 수가 총 4개이며, 전체 스캔 체인의 수도 30정도로 작다. 따라서 TAM=32이후의 경우들에 대해서는 모든 알고리즘들이 동일한 결과를 보여주고 있다.

하지만 P93791 이나 P22810과 같은 회로들의 경우

표 4-1. ITC'02 SOC 벤치 회로의 테스트 시간 (테스트 클럭) 비교
Table 4-1. Result of the Test Time for ITC'02 SOC Bench Circuits.

실험 회로	적용 방법	# of TAM lines (TAM bandwidth)						
		16	24	32	40	48	56	64
D695	본 논문	**37834**	**25727**	**19850**	**18564**	**14420**	**11341**	**10100**
	[12]	41949	28327	21423	17210	16403	13023	12327
	[13]	43447	30521	22955	18679	15982	13444	11520
	[14] CPLEX	42293	28305	21474	17612	14317	12466	10877
	[14] ECTPSol	41299	28288	20858	16799	14331	12148	10462
	[15]	43723	30317	23021	18459	15698	13415	11604
	[16]	43290	29023	21567	18799	17210	13488	12466
	[17]	46152	30777	22669	19551	16388	13877	11893
P22810	본 논문	**351441**	**245724**	**193451**	**148517**	**145417**	**123577**	**111383**
	[12]	462210	361571	312659	278359	268472	266800	260638
	[15]	452639	307780	246150	197293	167256	145417	136941
	[17]	541402	356039	294788	220674	203257	175946	157527
P34392	본 논문	**875499**	**622163**	583494	**544579**	**544579**	**544579**	**544579**
	[12]	998733	720858	591027	**544579**	**544579**	**544579**	**544579**
	[15]	1023820	759427	**544579**	**544579**	**544579**	**544579**	**544579**
	[17]	1191289	838643	568133	**544579**	**544579**	**544579**	**544579**
P93791	본 논문	**1585777**	**1080924**	**817846**	**657218**	**552047**	**539173**	**420187**
	[12]	1771720	1187990	887751	698583	599373	514688	460382
	[15]	1851135	1248795	975016	794020	627934	568436	511286
	[17]	2404341	1598829	1179795	1060369	717602	625506	491496

칩을 구성하는 코어의 수도 많고 스캔 체인의 수도 많은 큰 회로들이다. 이 회로들의 경우 실험 결과를 보면 TAM의 크기가 큰 경우와 작은 경우 모두 기존의 알고리즘들과 비교할 때, 더 적은 테스트 시간을 가지는 테스트 패턴을 구하고 있다. 즉 본 논문에서 제시한 알고리즘이 크고 복잡한 회로에 대해서 더 좋은 결과를 구하고 있다.

D695는 회로를 구성하는 코어들의 체인 구성 상태가 비교적 평형 상태를 유지하고 있다. 따라서 TAM을 증가하면서 얻을 수 있는 효율성이 다른 회로들에 비해

떨어진다. 하지만 실험결과를 보면 본 논문에서 제시한 알고리즘이sub-optimized 결과를 찾는데 있어서도 다른 방식들 보다 더 좋은 결과를 찾고 있다.

V. 결 론

본 논문에서는 SOC 테스트에서 가장 기본적인 스캔 기반 테스트에 대해서 알아보고 테스트 시간을 줄이기 위해서 테스트 래퍼 구성에서 TAM을 평형 상태로 만드는 방법을 제안하였다. 그리고 전체 SOC 테스트 구조를 결정하는 방법과 하드웨어 오버헤드를 줄이는 방법을 제안함으로써 기존의 방식들 보다 더 성능이 좋은 테스트 전략을 제시했다. 결과를 보면 대부분 기존의 방식들 보다 10~40% 정도 나은 성능을 보이고 있으며, 기존의 잘 알려진 알고리즘을 응용하기 때문에 실제 응용 또한 용이하다.

참 고 문 헌

[1] IEEE P1500 Standard for Embedded Core Test (http://grouper.ieee.org/groups/P1500).

[2] J. Aerts and E. J. Marinissen, "Scan chain design for test time reduction in core-based ICs", Pro-ceedings of International Test Conference, pp. 448-457, 1998.

[3] M. L. Bushnell and V. D. Agrawal, "Essentials of electronic testing for digital, memory, and mixed-signal VLSI circuits", Kluwer Academic Publ., ISBN 0-7923-7991-8.

[4] E. J. Marinissen, S. K. Goel and M. Lousberg, "Wrapper design for embedded core test", Procee-dings of International Test Conference, pp. 911-920, 2000.

[5] R. L. Graham, "Bounds on multiprocessing anomalies", SIAM Journal of Applied Mathe-matics, Volume 17, pp. 416-429, 1969.

[6] E. Larsson and Hideo Fujiwara, "Power constra-ined preemptive TAM scheduling", Proceedings of the Seventh IEEE European test Workshop, 2002.

[7] Wei Zou, S. M. Reddy, I. Pomeranz and Yu Huang, "SOC test scheduling using simulated annealing", VLSI Test Symposium, 2003. Procee-dings. 21st, pp. 325 -330, 27 April - 1 May 2003.

[8] P. Varma and B. Sandeep, "A structured test re-use methodology for systems on silicon", Proceedings of International Test Conference, pp. 294-302, 1998.

[9] ITC'02 (International Test Conference) SOC Benchmarks (http://www.extra.research.philips.com/itc02socbench.com/).

[10] E. G. Coffman Jr., M. R. Garey and D. S. Johnson, "An application of bin-packing to multiprocessor scheduling", SIAM Journal of Computing, Volu-mn 7, Number 1, pp. 1-17, 1978.

[11] C. Sunghoon, Y. Kim, Y. Shin, S. Song and S. Kang, "A new functional delay fault ATPG for embedded cores", Proceedings of the 4th Korea Test Conference, pp. 159-164, 2003.

[12] V. Iyengar, K. Chakrabarty and E. J. Marinissen, "Test wrapper and test access mechanism co-optimization for system-on-chip", Proceedings of International Test Conference (ITC02), pp. 1023-1032, 2001.

[13] S. Koranne, "On test planning for core-based SOCs", Proceedings of ECCO XIV, 2001.

[14] S. Koranne, "Design of reconfigurable access wrappers for embedded core based SOC test", Proceedings of the International Symposium on Quality Electronic Design (ISQED02), pp. 106-111, 2002.

[15] V. Iyengar, K. Chakrabarty and E. J. Marinissen, "On using rectangle packing for SOC wrapper/TAM co-optimization", Proceedings of VLSI Test Symposium, 2002. (VTS 2002), pp. 253-258, 2002.

[16] S. Koranne, "Formulating SOC test scheduling as a network transportation problem", Transac-tions on Computer-Aided Design of Integrated Circuits and Systems Volume: 21 Issue: 12, pp. 1517-1525, 2002.

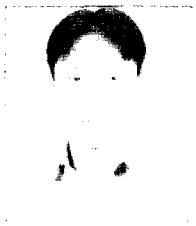
[17] S. Koranne and V. Iyengar, "On the use of k-tuples for SOC test schedule representation", Proceedings of International Test Conference (ITC02), pp. 539 -548, 2002.

저 자 소 개

강 길 영(정회원)
2001년 연세대학교 전기공학과 학사 졸업.
2003년 연세대학교 전기전자공학과 석사 졸업.
2004년 현재 삼성전자 반도체총괄 메모리사업부
<주관심분야: DFT, CAD, VLSI, Testing>

김 근 배(정회원)
2003년 연세대학교 전기공학과 학사 졸업.
2004년 연세대학교 전기전자공학과 석사 졸업.
2004년 현재 연세대학교 전기전자공학과 박사 과정
<주관심분야: On-line test, DFT, CAD, VLSI>

임 정 빈(정회원)
2003년 연세대학교 기계전자공학부 전기전자전공 학사 졸업.
2004년 현재 연세대학교 전기전자공학과 석사 과정
<주관심분야: DFT, Testing>



전 성 훈(정회원)
2002년 연세대학교 전기공학과 학사 졸업.
2004년 현재 연세대학교 전기전자공학과 석사 과정
<주관심분야: DFT, CAD>

강 성 호(정회원)
1986년 서울대학교 공대 제어계측공학과 학사 졸업
1988년 The university of Texas at Austin 전기 및 컴퓨터 공학과 석사 졸업
1992년 The university of Texas at Austin 전기 및 컴퓨터공학과 박사 졸업
미국 Schlumberger 연구원, Motorola 선임 연구원
현재 연세대학교 전기전자공학과 교수
<주관심분야: SoC 설계 및 SoC 테스트>