

# 스캔체인 레이아웃 거리를 고려한 Test Wrapper 설계

정 준 모

군산대학교 전자정보공학부

e-mail: jmjung@kunsan.ac.kr

## Efficient Test Wrapper Design Considering Layout Distance of Scan Chain

Jun Mo Jung

Dept of Electronic & Information, Kunsan National University

### 요 약

본 논문에서는 스캔 체인의 레이아웃 거리를 고려한 효율적인 Test Wrapper 설계 방식을 제안한다. SoC내의 스캔체인들을 테스트 하기 위해서는 외부 TAM line에 각 스캔체인들을 할당해야 한다. IP 내에 존재하는 스캔체인들은 스캔체인간 레이아웃 거리를 갖게 되며 이 거리가 클럭주기를 넘어가는 경우 체인의 타이밍 위반(Timing violation)이 발생할 수 있다. 본 논문에서는 타이밍 위반이 발생하지 않도록 체인간 거리를 고려하여 스캔체인을 할당하는 새로운 test wrapper 설계 방식을 제안하였다.

### 1. 서론

최근 개발되는 SoC(System-On-a-Chip)들의 집적도가 높아지고 각 IP에 접근하기가 어려워지고 있기 때문에 테스트 시간이 점점 길어지고 있다. 이에 따라 SoC내의 각 IP에 효율적으로 접근하기 위한 테스트 기법이 활발히 연구되고 있으며 표준화 그룹인 IEEE P1500 등에서 현재 개발중이다[1]. SoC내의 IP 코어로 직접 접근이 가능하도록 해주는 Wrapper는 TAM(Test Access Mechanism)과 IP를 연결해주는 핵심요소이다. 코어내부의 스캔체인과 코어의 입출력 터미널의 균형있는 연결을 위한 연구가 진행되어 왔다.

Wrapper 설계시에 고려되어야 할것은 전체 테스트 시간과 테스트 면적 오버헤드이다. 한정된 개수의 TAM line을 스캔체인과 연결할 때 테스트 시간이 최적화 되도록 테스트 스케줄링하는 연구가 진행되었다. TAM line의 최장 테스트 시간이 최소화 되도록 스캔체인을 할당하는 문제는 NP-Hard 한 문제로 분류되며 대표적 알고리즘이 LPT(Largest Processing Time) 기법으로 체인의 길이를 기준으로 체인들을 내림차순으로 정렬시킨후에 각 TAM

line에 순서대로 할당하는 방법이다[2]. 보다 향상된 알고리즘인 MULTIFIT 기법[3]과 LPT 기법을 조합한 Combine 기법이 제안되었다[4].이 기법은 LPT에서 구해진 최장 테스트 시간을 기준으로 한계치를 설정한 후에 이것을 초과하지 않는한 동일한 TAM line에 지속적으로 스캔체인을 할당하는 기법이다. 테스트 시간과 면적 오버헤드를 동시에 최적화 하는 새로운 기법이 제안되었다[5]. 이 논문에서는 LPT함수를 수행한 후, 무조건 FFD 함수를 수행하게 함으로써 보다 향상된 결과를 보여준다.

본 논문에서는 스캔 체인간 레이아웃 정보를 이용하여 타이밍 위반(Timing Violation)이 없는 효율적인 TAM 할당 알고리즘을 제안한다. 코어에 내장된 스캔체인들을 배치(Placement) 및 라우팅(Routing), 즉 레이아웃(Layout)하면 TAM line에 연결하고자 하는 스캔체인간에 거리(distance)가 존재하게 되며 거리로 인한 신호 지연시간이 시스템 클럭의 주기를 넘어가면 플립플롭의 셋업(setup) 및 홀드(Hold) 타임 위반(violation)이 발생한다. SoC의 집적도가 높아짐에 따라 거리로 인한 시간지연이 상대적으로 길어지게 되므로 이런 현상이 일어날 가능성이 점점 높아

지게 된다. 본 논문에서는 기존의 알고리즘에 타이밍 위반 조건을 추가하여 효율적인 테스트 Wrapper 설계 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 SoC 테스트의 구성요소 및 이전의 연구기법들에 대하여 기술하였으며 3장에서는 논문에서 제안하는 레이아웃 거리를 고려한 효율적인 TAM line 할당 알고리즘을 기술하였고 4장은 실험 및 결과, 5장에서는 결론을 기술하였다.

**2. SoC 테스트의 구성요소 및 이전의 연구기법들**

SoC 테스트의 3가지 필수 구성요소는 테스트 패턴 소스(Source)와 싱크(Sink), 테스트 접근장치(TAM: Test Access Mechanism) 및 Wrapper 이다. 소스는 코어의 테스트 패턴을 생성하며 싱크는 테스트 응답과 정상 응답을 비교한다. 테스트 접근장치는 IP 코어의 터미널과 TAM 라인을 연결해주는 스위치의 기능을 한다. TAM 라인의 수와 IP 코어의 터미널 수는 다를 수 있으며 한개의 TAM라인에 포함된 스캔체인들은 순차적으로 테스트되며 그림 1은 TAM에 할당된 스캔체인의 구성 예제를 나타낸 것이다.

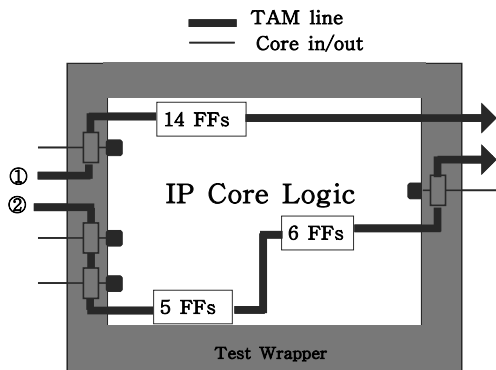


그림 1. Test Wrapper 설계 예

TAM 라인 ①은 14개의 플립플롭(F/F)들로 구성된 스캔체인으로 구성되고 wrapper cell 1개와 연결되어 총 15개로 연결되며, TAM 라인 ②는 5개, 6개 플립플롭들로 구성된 스캔체인을 포함하며 wrapper cell 3개를 거치므로 총 14로 연결되어 각 TAM 라인별로 동일한 시간으로 테스트가 병렬로 수행될 수 있다. 일반적인 테스트 Wrapper 설계기법은 이와같이 테스트 시간이 최적으로 스케줄링되도록 TAM 라인에 스캔체인을 할당한다.

**2.1. LPT, FFD, Combine 알고리즘**

LPT 알고리즘은 y개의 스캔체인을 m개의 TAM

라인에 할당하는 알고리즘이며, 모든 스캔체인을 체인 길이의 내림차순으로 정렬하고 우선 m개의 스캔체인을 각 TAM 라인에 한개씩 할당하고 나머지 스캔체인은 가장 적은 길이를 갖는 TAM 라인에 할당한다. 이와 같은 순서대로 모든 스캔체인을 할당한다. 리턴값은 TAM 라인의 최대길이이다.

Combine 기법은 LPT와 FFD 알고리즘을 결합하였다. FFD 알고리즘은 TAM 라인의 길이와 할당될 체인 길이의 합이 한계치 C보다 작으면 해당 체인을 할당하는 방식이다. Combine 기법은 LPT 알고리즘으로 한계치 C를 구한후에 FFD를 수행하는 방법이다.

각 알고리즘의 의사코드를 그림 2에 나타내었다.

**1) LPT 알고리즘**

```
// m: TAM 라인의 갯수
// y: scan 체인의 갯수
// L(Si): 스캔체인 Si의 길이
// Pi: TAM 라인
```

```
sort Si such that L(S1) >= L(S2) -- L(Sy);
for(i=1; i <= m; i++)
    Pi = Si;
for(i=m+1; i <= y; i++)
    select k ∈ {j | L(Pj) = min1 ≤ x ≤ m L(Px)};
    Pk = Pk ∪ Si;
return max1 ≤ x ≤ m L(Px);
```

**2) FFD(C) 알고리즘**

```
// C: 한계치
sort Si such that L(S1) >= L(S2) -- L(Sy);
j=1;
for(i=1; i <= y; i++)
    while(1)
        if((L(Pj) + L(Si)) <= C)
            Pj = Pj ∪ Si;
            continue;
        else
            j++;
return max {j | Pj ≠ 0};
```

그림 2. 기존 알고리즘의 Pseudo Code

### 3. 레이아웃 거리를 고려한 효율적인 TAM 라인 할당 알고리즘

레이아웃 거리는 회로를 P&R(Placement & Routing) 하게 되면 발생하게 되며, 이로 인해 신호의 지연시간이 생기게 된다. 이 지연시간이 시스템 클럭의 타이밍 조건에 만족한다면 시스템의 기능적 수행에는 문제가 없다. 하지만 칩의 집적도가 매우 높아지고 반도체 공정이 나노공정으로 바뀌면서 작은 신호의 지연시간에도 타이밍 문제가 발생할수 있다. 기존의 TAM 라인 할당 알고리즘에는 이러한 문제들이 전혀 고려되지 않고 있다. 본 논문에서는 기존의 할당 알고리즘에 레이아웃 거리를 적용하여 타이밍 문제가 없도록 하였다.

그림 3은 제안한 알고리즘의 Pseudo Code이다. 기존의 모든 알고리즘에 적용가능하지만 LPT 알고리즘에 적용하였다.

```

// m: TAM 라인의 갯수
// y: scan 체인의 갯수
// L(Si): 스캔체인 Si의 길이
// Pi: TAM 라인

sort Si such that L(S1)>=L(S2)--L(Sy);
for(i=1; i<=m; i++)
    Pi = Si;
for(i=m+1; i <= y; i++)
{
    select k ∈{j| L(Pj) = min1≤x≤m L(Px)};
    select Si with minimum layout distance
        Pk = Pk ∪ Si;
}
return max1≤x≤m L(Px);
    
```

그림 3. 제안한 알고리즘의 Pseudo Code

### 4. 실험 및 결과

제안한 알고리즘을 벤치마크 회로에 적용하여 실험하였다. 이 회로는 총 9개의 스캔체인으로 구성되었으며, 체인의 길이가 10인 체인은 2개, 길이가 8인 체인은 3개, 길이가 4인 체인은 2개이고 길이가 2인 체인도 2개이다. 각 체인간 상대적인 레이아웃 거리는 시스템 클럭을 기준으로 정규화하여 거리가 클럭

의 주기보다 긴경우를 1, 작은 경우를 0로 하여 시뮬레이션 하였다. 표 1의 A\_B는 A는 스캔체인의 길이이고 B는 그중 해당 스캔체인의 번호이다. 즉 10\_1은 체인이 길이가 10인 첫번째 스캔체인을 의미한다. 10\_2와 4\_1은 레이아웃 거리가 1이며 또한 8\_1과 8\_2의 레이아웃 거리도 1 임을 의미한다.

표 1. 실험 결과

	레이아웃 거리	기존의 알고리즘	제안된 알고리즘
10_2, 4_1	1	10_2, 4_1 연결시 실패	성공
8_1, 8_2	1	8_1, 8_2 연결시 실패시	성공

### 5. 결론

제안한 알고리즘은 레이아웃 거리를 고려한 TAM 라인 할당 방식으로 하였다. 시스템의 클럭주파수가 증가하고 집적도가 높아짐에 따라 레이아웃 거리를 고려한 방식은 매우 효율적으로 적용될 수 있다.

### 참고문헌

- [1] IEEE P1500 General Working Group website, "IEE P1500 Standards For Embedded Core Test, " <http://grouper.ieee.org/groups/1500>
- [2] R.L. Graham, "Bounds on Multiprocessing Anomalies," SIAM Journal of Applied Mathematics, Volume 17, pp.416-429, 1969.
- [3] E.G. Coffman Jr., M.R. Garey, D.S. Johnson, "An Application of Bin-Packing to Multiprocessor Scheduling," SIAM Journal of Computing, Volume 7, Number 1, pp.1-17, 1978.
- [4] Lee, C.Y., D. Massey, "Multiprocessor Scheduling: Combining LPT and Multifit," Discrete Applied Mathematics, Volume 20, pp.233-242, 1988.
- [5] 김문준, 박재홍, 최선화, 심재현, 장훈, "내장된 코어 테스트를 위한 효율적인 Wrapper 설계 기법", 테스트 학술대회, 2003.