

SoC 내의 효율적인 Test Wrapper 설계

정준모^{1*}

¹국립 군산대학교 전자공학과

Efficient Test Wrapper Design in SoC

Jun-Mo Jung^{1*}

¹Dept. of Electronic Engineering, Kunsan National University

요약 본 논문에서는 스캔 체인의 레이아웃 거리를 고려한 효율적인 Test Wrapper 설계 방식을 제안한다.

SoC내의 스캔체인들을 테스트하기 위해서는 외부 TAM 라인(line)에 각 스캔체인들을 할당해야 한다. IP 내에 존재하는 스캔체인들은 정상모드에서는 타이밍 위반(Timing Violation)이 발생하지 않도록 레이아웃이 되지만, 테스트 모드에서는 TAM 라인(line)과 연결되는 스캔체인들 간에 부가적인 레이아웃 거리를 갖게 되므로 스캔체인에서 타이밍 위반이 발생될 수 있다. 본 논문에서는 타이밍 위반이 발생하지 않도록 체인간 레이아웃거리를 고려하여 스캔체인을 할당하는 새로운 test wrapper 설계 방식을 제안하였다.

Abstract We present the efficient test wrapper design methodology considering the layout distance of scan chain. To test the scan chains in SoC, the scan chains must be assigned to external TAM(Test Access Mechanism) lines. The scan chains in IP were placed and routed without any timing violation at normal mode. However, in test mode, the scan chains have the additional layout distance after TAM line assignment, which can cause the timing violation of flip-flops in scan chains. This paper proposes a new test wrapper design considering layout distance of scan chains with timing violation free.

Key Words : SoC, Test Wrapper, Layout Distance

1. 서론

최근 개발되는 SoC(System-On-a-Chip)들의 집적도가 높아지고 각 IP(Intellectual Property)에 접근하기가 어려워지고 있기 때문에 테스트 시간이 점점 길어지고 있다. 이에 따라 SoC 내의 각 IP에 효율적으로 접근하기 위한 테스트 기법들이 활발히 연구되고 있으며 표준화 그룹인 IEEE P1500 등에서 현재 개발 및 표준화 작업을 진행하고 있다[1]. SoC내의 IP 코어로 직접 접근이 가능하도록 해주는 Wrapper는 TAM (Test Access Mechanism)과 IP를 연결해주는 핵심요소이다.

코어내부의 스캔체인과 코어의 입출력 터미널의 균형 있는 연결을 위한 연구가 진행되어 왔다.

Wrapper 설계시에 고려되어야 할 것은 전체 테스트

시간과 테스트 면적 오버헤드이다.

한정된 개수의 TAM 라인(line)에 스캔체인들을 연결하는 알고리즘에서는 테스트 시간을 최적화하는 테스트 스케줄링 연구가 진행되었다. TAM 라인의 최장 테스트 시간이 최소화 되도록 스캔체인을 할당하는 문제는 NP-Hard 한 문제로 분류되며, 대표적 알고리즘은 LPT(Largest Processing Time) 기법이다. 이 기법은 체인의 길이를 기준으로 체인들을 내림차순으로 정렬시킨 후에, 각 TAM 라인에 차례대로 체인들을 할당하는 방법이다[2]. 이 LPT 알고리즘을 보다 향상시킨 방법으로 MULTIFIT 기법이 제안되었으며[3], MULTIFIT 방법과 LPT 알고리즘을 조합한 Combine 기법이 제안되었다[4]. Combine 기법은 LPT에서 구해진 최장 테스트 시간을 이용하여 한계치를 설정한 후에, 이것을 초과하지 않는 범

본 연구는 군산대학교 정보통신기술연구소의 부분적인 지원으로 수행되었음.

*교신저자 : 정준모(jmjung@kunsan.ac.kr)

접수일 09년 02월 23일

수정일 (1차 09년 05월 14일, 2차 09년 06월 13일)

게재확정일 09년 06월 17일

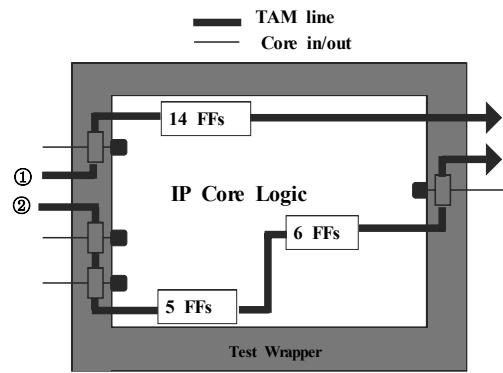
위내에서 동일한 TAM 라인에 지속적으로 스캔체인을 할당하는 기법이다. 또한, 테스트 시간과 면적 오버헤드를 동시에 최적화 하는 새로운 기법이 제안되었다[5]. 이 논문에서는 LPT 함수를 수행한 후, 무조건 FFD 함수를 수행하게 함으로써 보다 향상된 결과를 보여준다.

본 논문에서는 스캔 체인간 레이아웃 정보를 이용하여 타이밍 위반(Timing Violation)이 없는 효율적인 TAM 할당 알고리즘을 제안한다. 정상적인 기능적 모드에서는 코어에 내장된 스캔체인들이 배치(Placement) 및 라우팅(Routing), 즉 레이아웃(Layout)되면 타이밍 위반이 발생하지 않는다. 왜냐하면 레이아웃하는 동안 타이밍 위반에 대한 고려를 하기 때문이다. 그러나 테스트모드로 전환되어 체인들을 TAM 라인에 할당하게 되면 체인간에 새로운 신호 지연시간(delay time)이 추가로 발생하게 되고, 지연시간이 시스템 클럭의 주기를 넘어가면 플립플롭의 셋업(setup) 및 홀드(Hold) 타임 위반(violation)이 발생할 수 있다. SoC의 집적도가 높아짐에 따라 거리로 인한 시간지연이 상대적으로 길어지게 되고, 시스템 클럭의 주파수가 높아지면 이런 현상이 발생할 가능성이 점점 높아지게 된다.

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

2. SoC 테스트의 구성요소 및 이전의 연구방법

SoC 테스트의 3가지 필수 구성요소는 테스트 패턴 소스(Source)와 싱크(Sink), 테스트 접근장치(TAM) 및 Wrapper 이다. 소스에서는 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 TAM 라인에 포함된 스캔체인들은 순차적으로 테스트 된다. 그림 1은 스캔체인 3개를 2개의 TAM 라인에 연결하여 구성한 예제를 나타낸 것이다. 각 체인의 길이는 14, 6 및 5개이다.



[그림 1] Test Wrapper 설계 예

정상기능 모드(Functional Mode)에서는 각각의 체인들이 독립적인 기능으로 동작을 하며, 테스트 모드(Test Mode)에서는 외부 테스트 장비와 연결되는 TAM 라인에 체인들이 할당된다. TAM 라인을 통하여 테스트 입력을 인가하거나 테스트 응답을 출력하게 된다. 그림 1에 나타난바와 같이 TAM 라인은 2개(①, ②)가 존재한다. TAM 라인 ①은 14개의 플립플롭(F/F)들로 구성된 스캔체인으로 구성되고 wrapper cell 1개와 연결되어 총 15개로 연결된다. TAM 라인 ②는 5개와 6개 플립플롭들로 구성된 스캔체인을 포함하고 wrapper cell 3개를 거치므로 총 14개의 플립플롭으로 연결된다. ①과 ②는 별도의 TAM 라인이므로 각 TAM 라인별로 유사한 시간(플립플롭 14개 혹은 15개)으로 테스트가 병렬로 수행될 수 있다. 이와 같이 일반적으로 테스트 Wrapper 설계기법은 테스트 시간이 최적으로 스케줄링되도록 TAM 라인에 스캔체인을 할당한다.

2.1 LPT, FFD, Combine 알고리즘

LPT 알고리즘은 y개의 스캔체인을 m개의 TAM 라인에 효율적으로 할당하는 알고리즘이다. 모든 스캔체인을 체인 길이의 내림차순으로 정렬하고, 우선 m개의 스캔체인을 각 TAM 라인에 한개씩 할당하며, 나머지 스캔체인은 가장 길이가 작은 TAM 라인에 할당한다. 이와 같은 순서대로 모든 스캔체인을 할당한다. 리턴값은 TAM 라인의 최대길이이다.

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

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

각 알고리즘의 의사코드를 그림 2에 나타내었다.

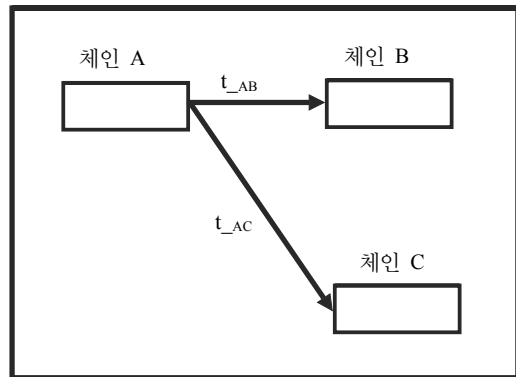
3. 레이아웃 거리를 고려한 효율적인 TAM 라인 할당 알고리즘

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. 정상 동작모드에서는 타이밍 위반이 발생하지 않도록 레

이아웃을 하게 된다. 하지만 테스트 모드에서는 스캔체인들이 TAM 라인에 할당되며, 이때 새로운 레이아웃 환경이 생성된다. 스캔체인들이 체인들과 서로 연결되면서 부가적인 레이아웃 거리를 갖게 된다. 이 부가적인 레이아웃 거리로 인하여 신호들의 지연시간이 길어지게 되며, 스캔체인들을 TAM 라인에 할당할 때 체인간의 레이아웃 거리를 고려하지 않는 경우에는, 레이아웃 거리로 인한 타이밍 위반이 발생할 수 있다. 기존의 TAM 라인 할당 알고리즘에는 이러한 문제들이 전혀 고려되지 않고 있다. 본 논문에서는 기존의 할당 알고리즘에 레이아웃 거리를 적용하여 타이밍 문제가 발생하지 않도록 효율적으로 Test Wrapper를 설계하였다.

그림 3의 예제회로에 레이아웃 거리에 대한 개념을 나타내었다.

스캔 체인이 3개(A,B,C)가 있으며 체인A와 연결될수 있는 대상 체인이 B와 C가 있다고 가정하였다. 체인 A의 종단 플립플롭과 체인 B와 C의 첫번째 플립플롭이 서로 연결되며 각 연결시에 레이아웃 거리가 발생하게 된다. 정상 기능동작에서는 문제없이 동작이 되어도 시스템이 테스트 모드로 변환되어 구성되면 새로운 플립플롭의 연결이 발생하는 것이다.



[그림 3] 레이아웃 거리에 대한 예제회로

이 예제회로의 클럭 주기를 T라 하고, 체인 A와 체인 B, 체인 A와 체인 C의 레이아웃 거리로 인해서 발생하는 시간을 각각 t_{AB} , t_{AC} 라고 한다면 시스템의 타이밍 안정성을 높이기 위해서는 t_{AB} , t_{AC} 가 주기 T보다는 작아야 한다. 왜냐하면 플립플롭간의 셋업(setup)/홀드(Hold) 시간을 포함한 플립플롭의 데이터 신호지연 시간이 시스템 클럭의 주기보다 크면 타이밍 위반(timing violation)을 일으키기 때문이다.

그림 3의 예제와 같이 임의의 스캔체인과 연결될수 있는 대상 체인의 레이아웃 거리를 계산한 후, 타이밍 위

반이 없는 대상 스캔체인을 선택하여 TAM 라인을 구성하는 새로운 알고리즘을 제안하였다.

그림 4는 제안한 알고리즘의 Pseudo Code이다. 연결 대상이 되는 모든 스캔체인간의 레이아웃 거리를 추출하고 그중에서 최소가 되는 레이아웃 거리를 구하고 타이밍 위반이 없으면 연결대상 체인으로 선택하게 된다.

$m, y, L(S_i), P_i$ 는 각각 TAM 라인의 갯수, scan 체인의 갯수, 스캔체인 S_i 의 길이 및 TAM 라인을 의미한다. 먼저, 스캔체인 S_i 의 길이를 기준으로 내림차순으로 정렬한다(①). 스캔체인을 순서대로 TAM 라인에 할당한다(②).

```

// 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); //①
for(i=1; i<=m; i++) //②
    Pi = Si;
for(i=m+1; i <= y; i++) //③
{
select k ∈{j| L(Pj) = min1≤x≤m L(Px)}; //④
select Si with minimum layout distance //⑤
if(timing violatio!) //⑥
    Pk = Pk ∪ Si;
}
return max1≤x≤m L(Px);
    
```

[그림 4] 제안한 알고리즘의 Pseudo Code

스캔체인이 최소인 것을 선택하고 타이밍 위반이 있는지 조사하고 타이밍 위반이 없으면 해당 스캔 체인을 TAM 라인에 할당한다.(④⑤⑥).

4. 실험 및 결과

제안한 알고리즘을 예제회로와 ITC 02 SoC 테스트 벤치마크 회로중 P93791의 모듈 6 코어에 적용하였다.

예제회로는 총 9개의 스캔체인으로 구성되었으며, 체인의 길이가 10인 체인이 2개, 길이가 8인 체인이 3개, 길이가 4인 체인이 2개이고 길이가 2인 체인도 2개이다. 각 체인간 상대적인 레이아웃 거리를 변수로 주어서 처리하였다. 상대적 레이아웃 거리는 시스템 클럭을 기준으로 정규화하였다. 레이아웃 거리가 클럭의 주기보다 긴 경우

(타이밍 위반이 발생하는 경우)를 1, 작은 경우를 0 (타이밍 위반이 없는 경우)으로 하였다. 표 1은 예제회로의 실험 결과이다. 체인 조건에 입력된 A_B의 A는 스캔체인의 길이이고 B는 A의 스캔체인중 B번째 스캔체인을 의미한다. 즉 10_1은 체인이 길이가 10인 첫번째 스캔체인을 의미한다. 10_2와 4_1은 레이아웃 거리가 1이며 또한 8_1과 8_2의 레이아웃 거리도 1임을 의미하고 그외의 모든 체인간 레이아웃 거리는 0으로 간주하여 시뮬레이션 하였다.

[표 1] 예제회로 실험 결과

체인 조건	상대적 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

표 1에 나타낸 바와 같이, 기존의 알고리즘은 대상 스캔체인의 타이밍 위반을 고려하지 않았기 때문에 오류가 발생하는 경우에도 무조건 체인연결을 시도하여 타이밍 오류를 발생시킨다.

제안된 알고리즘은 연결 대상 체인의 상대적 레이아웃 거리가 1인 경우에는 레이아웃 거리가 0인 다른 체인을 이용하여 TAM 라인 구성을 할 수가 있다.

P93791의 모듈 6 코어의 특성을 표 2에 나타내었다. 모듈 6 코어는 417개의 입력터미널, 324개의 출력 터미널 및 72개의 양방향 터미널을 가지고 있으며 길이가 500개인 스캔체인이 7개, 520의 길이를 갖는 체인이 30개 그리고 521개의 길이를 갖는 체인이 9개를 포함하고 있다.

[표 2] P93791의 모듈 6 코어의 특성

입출력 단자			스캔 체인	
입력	출력	양방향	길이	수
417	324	72	500	7
			520	30
			521	9

표 3은 ITC 02 SoC 테스트 벤치마크 회로중 P93791의 모듈 6 코어에 적용한 실험결과이다.

[표 3] P93791의 모듈 6 코어의 실험 결과

COMBINE		제한한 방법		
Available TAM line	Utilized TAM line	Available TAM line	Utilized TAM line	Longest Wrapper Scan Chain
15	N/A	15	15	2080
16-19	N/A	16-19	16	2060
20-21	N/A	20-21	20	1560

기존의 Combine 기법과 제한한 방법을 비교하였으며 이용 가능한 TAM 라인은 15에서 21개로 제한하였다. 또한 임의의 TAM 라인에 레이아웃 거리를 1로 할당하여 시뮬레이션 하였다. COMBINE 논문의 알고리즘은 모든 경우에 TAM 라인을 구성하지 못하였으므로 사용된 TAM 라인의 갯수가 N/A(Non-Available)이며, 제한한 방법을 적용하면 모든 경우에 구성이 가능하였다. 가장 긴 Wrapper 스캔 체인의 길이는 이용 가능한 TAM 라인의 갯수가 15인 경우에는 2080, 16-19인 경우에는 2060 그리고 20-21인 경우에는 1560 이었다.

5. 결론

본 논문에서는 타이밍 위반이 발생하지 않는 새로운 방식의 TAM 라인 할당 방식을 제안하였다. 스캔체인들의 레이아웃 거리를 체인내 플립플롭의 시간적 지연으로 모델링한 후, 플립플롭의 셋업/홀드 타임의 위반이 발생하지 않도록 각 스캔체인들을 TAM 라인에 할당하였다. 제안한 방식을 P93791의 모듈 6 코어에 적용한 결과 기존의 알고리즘에서는 타이밍 오류가 발생하지만 제안한 방식을 적용하면 오류가 발생하지 않았다. 향후, 본 논문에서 제안한 방식을 새로운 형태의 SoC 코어 모듈에도 적용하여 효율성을 검증하도록 하겠으며, 시스템의 클럭 주파수가 증가하고 집적도가 높아짐에 따라 레이아웃 거리를 고려한 제안방식이 매우 효율적으로 적용될 수 있을 것으로 기대한다.

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.

정 준 모(Jun-Mo Jung)

[중심회원]



- 1987년 2월 : 한양대학교 전자공학과 (공학석사)
- 2004년 2월 : 한양대학교 전자공학과 (공학박사)
- 1989년 2월 ~ 1996년 3월 : 삼성전자 ASIC 센터
- 1996년 4월 ~ 2004년 2월 : 김포대학 전자정보계열 교수
- 2004년 3월 ~ 2005년 3월 : 한양사이버대학교 컴퓨터공학과 조교수
- 2005년 4월 ~ 현재 : 군산대학교 전자공학과 부교수

<관심분야>

VLSI Design, SoC Design, SoC Test & Verification, Test Scheduling