

테스트 핀 축소에 의한 저비용 SoC 테스트 구조

Reduced Pin-Count Test Architecture for Low-Cost SoC Test

이현빈, 김병진, 김진규, 권지연, 박성주

한양대학교 컴퓨터 공학과

{bean, bjkim, jkkim, jiyeon, parksj}@mslab.hanyang.ac.kr

Abstract

In this paper a reduced pin count SoC test architecture using IEEE 1149.1 and IEEE 1500 wrapper is presented. By using only a small number of test pins low cost automated test equipments (ATEs) can be efficiently utilized to SoC test cost. Experimental results show that the SoC test time can be significantly reduced by performing multi-site test.

I. 서론

재사용 가능한 IP (Intellectual Property) 모듈을 사용함으로써 설계 시간은 단축되었지만, System-on-a-chip (SoC)의 복잡도가 늘어나고 동작 속도가 빨라지면서 테스트의 비중은 점점 더 높아졌다. 따라서, IP 설계자와 SoC 설계자간의 테스트 비용을 줄이기 위한 노력이 요구되었다. 테스트 패턴 압축 (test data compression and response compaction), 핀 축소 테스트 (reduced pin count test (RPCT)), 온칩 딜레이 테스트 클럭 생성기, 테스트 접근 메커니즘 최적화, multi-site test 등을 통하여 저가의 ATE를 사용하거나 테스트 시간을 단축 시킴으로써 테스트 비용을 줄이는 연구가 이루어지고 있다[1-7].

본 논문에서는 IEEE 1149.1과 IEEE 1500 래퍼를 이용한 RPCT 구조를 제시하고 테스트 시간을 분석한다. 테스트 핀 수를 줄여 multi-site 테스트를 수행 함으로써 테스트 효율을 높일 수 있음을 보여준다. 2장에서 IEEE 1149.1을 통하여 IEEE 1500 wrapper를 제어하는 RPCT architecture를 제시하고, 3장에서 테스트 시간을 분석한다. 4장에서 실험 결과를 제시하고 5장에서 결론을 맺는다.

II. IEEE 1149.1과 IEEE 1500 래퍼를 이용한 RPCT 구조

그림 1은 IEEE 1149.1와 IEEE 1500 래퍼로 구성된 모듈러 SoC 테스트를 위한 구조이다[8]. WSI와 WSO를 이용한 직렬 테스트 도메인뿐만 아니라 WPI와 WPO를 통한 병렬 테스트 도메인도 포함하고 있다. 기존 연구에서도 기존에 많이 사용되었던 IEEE 1149.1 TAP을 주로 활용했으며[9-11], IEEE 1500 표준에서도 IEEE 1149.1 TAP을 통한 WSC 접근 로직을 제시하고 있다[12].

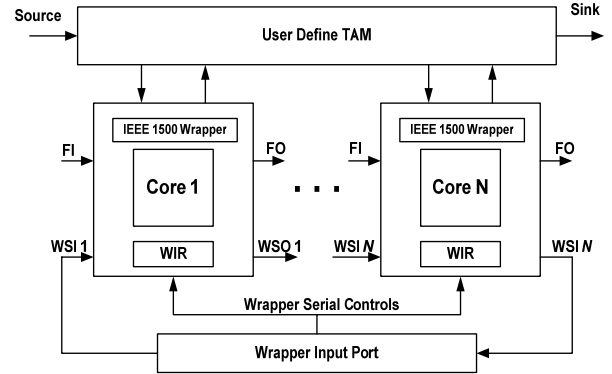


그림 1. IEEE 1500 래퍼를 통한 SoC 테스트 구조

본 논문에서는, WPP는 제외하고 WSP만을 다룬다. 모듈의 스캔 체인을 IEEE 1500 래퍼의 래퍼 레지스터와 직렬로 연결하고 (입력 WBR-스캔 체인-출력 WBR), Scan Enable 신호를 WSC의 ShiftWR과 연결하고, 간단한 클럭 게이팅 로직을 구현하여 IEEE 1500 래퍼와 스캔 체인에서 동시에 shift, update, capture 가 수행되도록 구성함으로써 각 모듈의 스캔 테스트 신호를 SoC의 외부로 빼지 않고 5개의 IEEE 1149.1 TAP 제어 핀만으로 SoC 내의 모든 모듈 테스트가 가능하다.

III. 제안한 RPCT 구조의 테스트 시간 분석

여러 개의 모듈로 이루어진 SoC 내의 한 모듈 m 에 대해서 테스트 패턴의 수를 p_m , 입력의 수를 i_m , 출력의 수를 o_m , 양방향 단자의 수를 b_m , 스캔 플립플롭의 수를 f_m 이라고 하면, 테스트 패턴 입력 비트수 $ts_m = i_m + b_m + f_m$ 이고 과 테스트 결과 출력 비트수 $tr_m = o_m + b_m + f_m$ 이 된다[5]. 기본적인 스캔 테스트 절차 (Scan In \rightarrow PI Application/Capture \rightarrow Scan Out)에서 현재 테스트 결과의 관측과 다음 테스트 패턴의 인가가 동시에 이루어지며, PI Application/Capture는 테스트 패턴 수만큼 수행하게 된다. 따라서, TAM (Test Access Mechanism)의 폭을 W_{TAM} 이라고 하면, 모듈 m 의 테스트 시간의 하계 (lower bound) LB_m 은 다음과 같다[5].

$$LB_m = \left\lceil \frac{\max(ts_m, tr_m) \cdot p_m + \min(ts_m, tr_m)}{W_{TAM}} \right\rceil + p_m$$

(1)

$$t_{ji} = \sum_{m=1}^{|M|} sft_m - o_{|M|} \quad (4)$$

본 논문에서는 SoC내의 모든 모듈을 하나의 TAM에 연결하고, 직렬 테스트 스케줄을 사용하는 경우와 병렬 테스트 스케줄을 사용하는 경우에 대해서 각각 테스트 시간을 분석하고 실험을 통하여 평가한다.

WIR 및 Bypass 레지스터를 통한 연결 설정 및 우선, 모듈 하나의 테스트 시간을 분석하면, $W_{TAM}=1$ 이고, Scan Enable='1'인 구간을 Shift 구간, Scan Enable='0'인 구간을 Capture 구간이라고 하면, Capture 구간의 길이가 4 TCKs이므로, 제한한 방식의 하나의 모듈의 테스트 시간 LTT_m 는 다음과 같다.

$$LTT_m = \max(ts_m, tr_m)p_m + \min(ts_m, tr_m) + 4p_m \quad (2)$$

1. Serial Schedule

직렬 테스트 스케줄을 사용하면, TAP 제어기를 통하여 테스트 할 모듈만 INTEST 모드로 설정하고 다른 모듈들은 모두 BYPASS로 설정하여 모듈 하나씩 순차적으로 테스트 한다. 따라서, 모듈의 총 집합을 M 이라고 할 때, 직렬 테스트 스케줄을 사용할 경우의 SoC 전체 테스트 시간의 하계 LTT_{SoC}^{ss} 는 각 모듈의 테스트 시간의 합이 되므로 다음과 같다.

$$LTT_{SoC}^{ss} = \sum_{m=1}^{|M|} LTT_m \quad (3)$$

2. Parallel Schedule

병렬 테스트 스케줄을 사용하면, TAP 제어기를 통하여 모든 모듈을 INTEST 모드로 설정하여 병렬로 테스트를 시작하고 테스트 패턴을 모두 소모한 모듈은 다른 모듈의 테스트가 끝날 때까지 BYPASS 모드를 유지하도록 한다. 따라서, 입출력 개수나 스캔체인 길이와는 상관없이 테스트 패턴수가 많은 모듈일수록 테스트가 늦게 끝나고, 테스트 패턴의 수가 같은 모듈들은 동시에 테스트가 끝나게 된다. 테스트 시간 분석을 위해서 $p_1 \leq p_2 \leq \dots \leq p_{|M|}$ 라고 가정하고, 테스트중인 모든 모듈에 일렬로 인가 및 관측해야 하는 테스트 패턴을 하나의 SoC 테스트 패턴이라고 하면, 총 $p_{|M|}$ 개의 테스트 패턴을 인가하고 관측해야 한다. 한 모듈의 총 패턴 쉬프트 수 $sft_m = i_m + b_m + f_m + o_m$ 라고 정의하고, 테스트 수행 시 고려해야 할 3 가지 경우와 그에 따라 소모되는 시간을 분석하면 다음과 같다.

Case1: 첫 번째 테스트 패턴 인가

결과 관측 없이 모든 모듈의 입력 패턴만 인가하면 된다. 따라서, core $|M|$ 의 출력셀에 적재할 필요는 없으므로 첫 번째 테스트 패턴을 인가하는데 걸리는 시간 $t_{i/o}$ 는 다음과 같다.

Case2: 마지막 테스트 결과 관측

자연수 k 에 대해서, 테스트 패턴의 수가 $p_{|M|}$ 개인 모듈이 k 개 이면, 즉, 자연수 h 에 대해서, $p_h = p_{h+1} = \dots = p_{|M|}$ 이고 $|M| - h + 1 = k$ 라면, k 개의 모듈은 동시에 테스트가 끝나고 이와 더불어 SoC 테스트도 종료된다. k 개의 모듈의 결과만 관측하면 되므로 core h 의 입력셀은 쉬프트 하지 않아도 된다. 따라서, 마지막 테스트 결과를 관측하는데 걸리는 시간 t_{lo} 는 다음과 같다.

$$t_{lo} = \sum_{m=h}^{|M|} sft_m - i_h \quad (5)$$

Case3: 중간 테스트 패턴 인가/결과 관측 및 마지막 테스트 결과 관측

현재의 테스트 결과 관측은 다음 테스트 패턴 인가와 동시에 수행된다. $|M|$ 이하인 자연수 j 에 대하여, core j 부터 core $|M|$ 까지 테스트 중일 때, 하나의 테스트 패턴 인가에 필요한 시간은 $\sum_{m=j}^{|M|} sft_m - o_{|M|}$ 이고, 하나의 테스트 결과 관측에 필요한 시간은 $\sum_{m=j}^{|M|} sft_m - i_j$ 이다. 따라서, 하나의 테스트 결과 관측과 하나의 테스트 패턴을 인가를 동시에 하는데 걸리는 시간은 $\sum_{m=j}^{|M|} sft_m - \min(i_j, o_{|M|})$ 이 된다. core j 부터 core $|M|$ 까지 테스트는 항상 core $(j+1)$ 부터 core $|M|$ 까지의 테스트를 포함하고 p_j 번째 테스트 결과 관측 후에 core j 의 테스트는 끝난다. 테스트 패턴의 수가 같은 모듈들은 동시에 테스트가 종료되므로 테스트 시간이 중복으로 계산되지 않아야 한다. 따라서, Case 2에서 정의한 바와 같이 p_h 번째 패턴이 마지막 패턴이라고 하면, 첫 번째 테스트 패턴 인가와 마지막 패턴 결과 관측을 제외한 모든 중간 테스트 패턴 인가/결과 관측 하는데 걸리는 시간 $t_{i/o}$ 는 다음과 같다.

$$\begin{aligned} t_{i/o} = & p_1 \cdot \left(\sum_{m=1}^{|M|} sft_m - \min(i_1, o_{|M|}) \right) \\ & + (p_2 - p_1) \cdot \left(\sum_{m=2}^{|M|} sft_m - \min(i_2, o_{|M|}) \right) \\ & \vdots \\ & + ((p_h - 1) - p_{h-1}) \cdot \left(\sum_{m=h}^{|M|} sft_m - \min(i_h, o_{|M|}) \right) \end{aligned} \quad (6)$$

Table 1. Multi-site 테스트에 의한 테스트 처리량

SoC Benchmarks	TR-Architecture [5]				Proposed			
	W_{opt}^1	LB_T^2	T_{TR}^3	D_{th}^4	LB_{pr}^5	T_{pr}^6	D_{th}	Improved D_{th} (%)
u226	48	5333	7999	25295	64653459	64653873	26919	6.4
d281	40	3926	3926	30388	4052534	4052800	241489	694.7
d695	64	10247	11033	20218	662343	662843	435810	2055.6
h953	16	119357	119357	78487	1142424	1142760	391240	398.5
g1023	40	14794	14794	30296	493737	494661	453931	1398.3
f2126	24	335334	335334	46328	5174665	5174769	210435	354.2
q12710	16	2222349	2222349	49893	24442515	24442619	65588	31.5
p22810	64	104868	133405	19547	6834979	6838633	176731	804.1
p34392	40	544579	544579	26376	15193256	15195056	97945	271.3
p93791	64	436673	455738	17974	27865575	27869991	58434	225.1

¹ W_{opt} : Optimized TAM width; ² LB_T : Lower Bound of testing time (# of test clocks); ³ T_{TR} : Testing time of TR-Architecture (# of test clocks); ⁴ D_{th} : # of devices tested per hour (=3600n/t_{ts}, n: # of devices, t: testing time(s)); ⁵ $LB_{pr} = \min(LTT_{SoC}^{SS}, LTT_{SoC}^{PS})$; ⁶ T_{pr} : Testing time of proposed method (# of test clocks).

식 (6)에서 마지막 항의 ($P_h - 1$)은 ($P_h - 1$) 번째 결과 관측까지만 다음 테스트 패턴 인가와 동시에 수행됨을 의미한다. P_h 번째 결과 관측 시에는 더 이상 인가할 테스트 패턴이 없으며 Case 2에서 계산된다. $P_0 = 0$ 으로 놓고 (6)을 다시 정리하면 다음과 같다.

$$t_{i/o} = \sum_{j=1}^h (p_j - p_{j-1}) \cdot \left(\sum_{m=j}^{|M|} sft_m - \min(i_j, o_{|M|}) \right) - \left(\sum_{m=h}^{|M|} sft_m - \min(i_h, o_{|M|}) \right) \quad (7)$$

최종적으로, 위의 두 가지 경우와 캡처하는데 걸리는 시간 $4p_{|M|}$ 을 모두 합함으로써 병렬 스케줄을 사용할 경우의 SoC 테스트 시간의 하계 LTT_{SoC}^{PS} 를 다음과 같이 구할 수 있다.

$$LTT_{SoC}^{PS} = t_{fi} + t_{i/o} + t_{lo} + 4p_{|M|} = \sum_{m=1}^{|M|} sft_m + \sum_{j=1}^h (p_j - p_{j-1}) \cdot \left(\sum_{m=j}^{|M|} sft_m - \min(i_j, o_{|M|}) \right) + \min(i_h, o_{|M|}) - i_h - o_{|M|} + 4p_{|M|} \quad (8)$$

IV. 실험

제시한 방식의 효율성을 평가하기 위해 ITC'02 SoC 테스트 벤치마크 회로 [13]를 사용하고 모든 래퍼의 WIR의 크기는 4 비트로 설계하여 테스트 시간을 측정하였다. 테스트 시간과 하나의 SoC를 테스트 하는데 필요한 핀 수를 기반으로, multi-site 테스트를 통한 단위 시간당 테스트 처리량을 분석하여 논문[5]의 TAM 최적화 방식과 비교하였다. Table 1에서, 논문 [14]에 제시된 가정을 바탕으로, ATE 채널의 수, index time, contact test time, 테스트 클럭 주파수를 각각 512, 0.7s, 10ms, 5 MHz로 가정하고 D_{th} 를 비교하였다. multi-test를 수행함으로써 단위 시간당 테스트 가능한 총 SoC의 수 D_{th} 를 분석한 결과 평균 511.6% 향상 시킬 수 있었다.

V. 결론

본 논문에서는, 모듈러 SoC 테스트를 위해 IEEE 1149.1과 IEEE 1500 래퍼를 이용한 핀수 축소 테스트 구조를 제시하고 테스트 시간을 분석하였다. multi-site 테스트를 수행함으로써 기존의 TAM방식에 비해 테스트 처리량을 향상시켰다. 제안한 방법을 사용함으로써 SoC의 핀 수를 줄이고 테스트 시간을 향상시킴으로써 테스트 비용을 줄일 수 있다.

Acknowledgement

본 연구보고서는 정보통신부의 출연금 등으로 수행한 정보통신연구개발사업의 연구결과입니다.

참고문헌

- [1] S. Hwang and J. A. Abraham, "Test Data Compression and Test Time Reduction Using an Embedded Microprocessor," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 11, pp. 853-862, Oct. 2003.
- [2] H. Vranken, et al, "Enhanced Reduced Pin-Count Test for Full-Scan Design," Proceedings of IEEE International Test Conference, pp. 738-747, 2001.
- [3] J. Jahangiri, N. Mukherjee, C. Wu-Tung, S. Mahadevan, and R. Press, "Achieving High Test Quality with Reduced Pin Count Testing," Proceedings of IEEE Asian Test Symposium, pp. 312-317, Dec. 2005.
- [4] M. Beck, O. Barondeau, M. Kaibel, F. Poehl, L.ub Xijiang and R. Press, "Logic Design for On-Chip Test Clock Generation Implementation Details and Impact on Delay Test Quality," Proceedings of the Design, Automation and Test in Europe, 2005.
- [5] S. K. Goel and E. J. Marinissen, "Effective and Efficient Test Architecture Design for

- SOCs," Proceedings of IEEE International Test Conference, pp. 529–538, Oct. 2002.
- [6] S. K. Goel and E. J. Marinissen, "On-Chip Test Infrastructure Design for Optimal Multi-Site Testing of System Chips," Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, Vol. 1, pp 44–49, 2005.
 - [7] H. Hashempour, F. J. Meyer, and F. Lombardi, "Analysis and Evaluation of Multisite Testing for VLSI," IEEE Transactions on Instrumentation and Measurement, Vol. 54, pp. 1770–1778, Oct. 2005.
 - [8] T. Waayers, E. J. Marinissen and M. Lousberg, "IEEE Std 1500 Compliant Infrastructure for Modular SOC Testing," Proceedings of Asian Test Symposium, pp 450, Dec. 2005.
 - [9] L. Whetsel, "Inevitable Use of TAP Domains in SoCs," Proceedings of IEEE International Test Conference, pp. 1191, Oct. 2002.
 - [10] Y. Zorian, A. Yessayan, "IEEE 1500 utilization in SOC design and test," Proceedings of IEEE International Test Conference, Nov. 2005.
 - [11] M. Schrader, R. McConnell, "SoC Design and Test Considerations," Design Automation and Test in Europe Conference and Exhibition, pp. 202–207, 2003.
 - [12] IEEE Computer Society, "IEEE Standard Testability Method for Embedded Core-based Integrated Circuits," Aug. 2005.
 - [13] E. J. Marinissen, V. Iyengar, and K. Chakrabarty, ITC' 02 SOC Test Benchmarks. Available:<http://www.hitech-projects.com/itc02socbenchm/>
 - [14] S. K. Goel and E. J. Marinissen, "On-Chip Test Infrastructure Design for Optimal Multi-Site Testing of System Chips," Proceedings of Design, Automation and Test in Europe, pp 44–49, 2005.