논문 2004-41SD-3-8

SOC 테스트를 위한 Wrapper 설계 기법

(An Efficient Wrapper Design for SOC Testing)

최 선 화*, 김 문 준*, 장 훈**

(SunHwa Choi, MoonJoon Kim, and Hoon Chang)

요 약

최근 하나의 칩에 여러 개의 코어들로 구성된 SOC(System on Chip) 테스트 비용의 증가로 인해 SOC 테스트에 있어서 재사용 방법론과 효율적인 테스트 방법의 중요성이 더욱 커지게 되었다. SOC 테스트의 일반적인 문제는 TAM(Test Access Mechanism)의 구조 설계와 테스트 코어 wrapper의 최적화, 테스트 스케줄링이 있다. 이러한 SOC 테스트의 목표는 테스트 시간과 하드웨어 오버헤드의 최소화이다. 이를 위해서 코어 내부의 스캔 체인과 입출력을 보다 균형 있게 배분하여 더 적은 테스트 시간과 TAM 너비를 사용하도록 테스트 시간과 하드웨어 오버헤드를 동시에 고려하여 설계 하는 것이 중요하다. 본 논문에서는 SOC 테스트를 위한 비용을 줄일 수 있는 코어 테스트 wrapper 설계 기법을 제안한다. 본 논문의 제안 기법은 기존의 기법들의 장점을 취하고 단점을 보완함으로써 보다 적은 테스트 시간과 하드웨어 오버헤드를 가진다. 이를 입증하기 위해서 ITC'02 SOC 테스트 벤치마크 회로를 이용하여 실험을 하였다.

Abstract

The SOC(System on Chip) testing has required the core re-use methodology and the efficiency of test method because of increase of its cost. The goal of SOC testing is to minimize the testing time, area overhead, and power consumption during testing. Prior research has concentrated on only one aspect of the test core wrapper design problem at a test time. Our research is concentrated on optimization of test time and area overhead for the core test wrapper, which is one of the important elements for SOC test architecture. In this paper, we propose an efficient wrapper design algorithm that improves on earlier approaches by also reducing the TAM(Test Access Mechanism) width required to achieve these lower testing times.

Keyword : SOC testing, Wrapper, LPT, FFD

I. 서 론

최근 하나의 칩에 여러 개의 코어들로 구성된 SOC(System on Chip)의 복잡도가 증가함에 따라 SOC 테스트의 비용도 점점 증가하고 있다. 이로 인해 SOC 테스트에 있어서 재사용 방법론과 테스트 방법의 효율성이 더욱 중요하게 되었다. SOC는 여러 개의 코어들로 이루어지는데, 코어는 검증된 미리 설계된 모듈이다. SOC 설계자는 코어 제공자에게 필요한 코어를 제공 받아서 SOC를 설계한다[1, 2, 3]. SOC의 모듈러 설계는 재사용 설계 기법을 가능하게 함으로써 칩의 제품 출시 기간을 단축시켰다[4, 5, 6].

SOC 테스트의 비용을 줄이기 위해서 SOC 테스트 기법에도 이러한 모듈러 설계 기법을 적용하게 되었다. 코어 제공자가 SOC 설계자에게 코어와 코어의 테스트 패턴 셋(set)을 동시에 제공하는 것이 그 예이다. 제공된 패턴 셋을 이용하여 외부에서 테스트하기 위해서는 SOC에 내장된 각각의 코어에 접근할 수 있어야 한다. TAM(Test Access Mechanism)과 테스트 wrapper는 SOC 테스트 접근 구조의 주요 요소이다. 1999년에 ITRS(International Technology Roadmap for Semiconductors)에서 이러한 SOC 테스트 접근 구조를 정의했으며[7], SOC 테스트 표준화 그룹인 IEEE P1500 등에

* 정회원, 숭실대학교 컴퓨터학과
(Department of Computing, Graduate School, Soongsil University)
** 정회원, 숭실대학교 컴퓨터학부
(School of Computing, Soongsil University)
※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음
접수일자 : 2003년12월4일, 수정완료일 : 2004년3월4일

서 SOC 테스트 접근 기법을 개발하고 있는 중이다[8, 9].

SOC 테스트의 일반적인 문제는 TAM 구조 설계와 코어 wrapper 최적화, 테스트 스케줄링 등이 있다. 이러한 SOC 테스트의 목표는 테스트 시간과 하드웨어 오버헤드의 최소화이다. 이는 코어 내부의 스캔 체인과 입출력을 보다 균형있게 배분하여 더 적은 테스트 시간과 TAM 너비를 사용하도록 테스트 시간과 하드웨어 오버헤드를 동시에 고려한 설계 기법이다. 본 논문에서 제안하는 코어 테스트 wrapper 설계 기법은 테스트 시간과 하드웨어 오버헤드를 동시에 고려하면서 기존의 기법보다 개선된 것이다. 이러한 코어의 최적화를 위한 테스트 스케줄링 알고리즘이 최근 집중적으로 연구되고 있다[10, 11].

본 논문은 다음과 같이 구성된다. Ⅱ장에서 기존의 Design_wrapper 설계 기법에 관하여 설명한다. Ⅲ장에서는 본 논문에서 제안하는 보다 향상된 코어 테스트 wrapper 설계 알고리즘을 설명한다. Ⅳ장에서 ITC'02 SOC 테스트 벤치마크 회로 중에서 p93791 회로를 이용한 실험 결과를 보인다. 마지막으로 Ⅴ장에서 결론을 맺는다.

Ⅱ. Design_wrapper 설계 기법

SOC 테스트 코어 wrapper의 효율적인 설계를 위해서는, 우선 코어 내부의 스캔 체인들을 주어진 TAM 라인에 균등하게 분배한 후 입출력 터미널들을 할당해야 한다. TAM 라인의 최장 테스트 시간이 최소화되도록 스캔 체인들을 할당하는 문제는 기존의 멀티프로세서 스케쥴링 문제, 빈 할당(bin-packing) 문제와 동일하며, 이는 NP-난해(NP-hard) 문제로 분류된다[10]. 이러한 문제들을 위해 개발된 대표적인 알고리즘인 LPT (Largest Processing Time) 기법[12]과 MULTIFIT 기법[13]을 조합한 알고리즘이 1988년에 개발되고 널리 사용되었다[14]. 이전 기법들은 TAM과 wrapper 설계의 연구에서 시간 단축이라는 한 가지 면에 집중해서 연구 했으나, Design_wrapper 기법은 시간뿐만 아니라 TAM의 설계와 wrapper의 최적화까지 고려한 방법이다. 즉, 더 적은 테스트 시간과 더 적은 TAM 너비를 사용하도록 테스트 시간과 하드웨어 오버헤드를 동시에 고려한 알고리즘이다. 이러한 Design_wrapper 알고리즘[11]은 이전의 코어 wrapper와 TAM 설계 기법의 한계를 극복한 기법이라고 할 수 있다. 본 논문에서 제안하는 새로운 테스트 코어 wrapper 설계 기법은 테스트 시간과 하드웨어 오버헤드의 감소를 동시에 고려하면서 Design_wrapper 알고리즘보다 향상된 기법이다. Design_wrapper 알고리즘의 의사코드는 그림 1과 같다. 단, y개의 코어 내부 스캔 체인의 집합 $S = \{S_1, S_2, \ldots, S_y\}$라고 하며 스캔 체인 S_i의 길이를 $Length(S_i)$로 표현한다. 주어진 m비트 TAM 라인의 집합 $TL = \{TL_1, TL_2, \ldots, TL_m\}$라고 하며 TL_i의 길이를 $Length(TL_i)$로 표현한다.

```
Design_wrapper
1 : do S descending sort
2 : do TL_j = NULL for all j
3 : for i ← 1 to y
4 :   do select
5 :     k∈{j|Length(TL_j) = min_{1≤x≤m} Length(S_max)
                          - (Length(S_i)+Length(TL_x))}
6 :   TL_k ← TL_k U {S_i}
7 :   if no such TL
8 :     then
9 :     if no available TL
10 :      then
11 :      do select k ∈ {j | Length(TL_j)
                          = min_{1≤x≤m} Length(TL_x)}
12 :      TL_k ← TL_k U {S_i}
13 :    else
14 :      then
15 :      do TL_used+1 ← TL_used+1 U {S_i}
16 :    end if
17 :  end if
18 : return the solution
```

그림 1. Design_wrapper 알고리즘
Fig. 1. Algorithm for Design_wrapper.

Design_wrapper 알고리즘은 우선 할당할 스캔 체인들을 길이의 내림차순으로 정렬한다. 그리고 스캔 체인이 할당되었을 때 해당 TAM 라인의 길이와 현재의 최장 TAM 라인의 길이와의 차이가 최소화되는 TAM 라인을 검색하여 해당 스캔 체인을 할당한다. 어느 TAM 라인에 할당해도 현재의 최장 TAM라인의 길이를 초과할 경우에는 새로운 TAM 라인에 할당한다. 단, 이러한 과정으로 스캔 체인들을 TAM 라인에 할당하다가 더 이상 주어진 TAM 라인이 없을 때에는 사용한 TAM 라인 중에서 최장 TAM 라인의 길이가 최소인 TAM 라인에 해당 스캔 체인을 할당한다.

Ⅲ. 제안된 코어 테스트 Wrapper 설계 기법

본 논문에서 제안하는 코어 테스트 wrapper 설계 기법은 COMBINE 기법[10]에 기초를 둔 알고리즘이다. 시

간 감소라는 측면에 중점을 둔 이전의 기법과 달리 이 알고리즘은 시간의 감소도 고려하면서 하드웨어 오버헤드 감소에 좀 더 큰 비중을 두고 있는 알고리즘이다. 본 논문에서 제안하는 코어 테스트 wrapper 알고리즘의 의사코드는 그림 2와 같다. 단, y개의 코어 내부 스캔 체인의 집합 $S = \{S_1, S_2, \ldots, S_y\}$라고 하며 스캔 체인 S_i의 길이를 $Length(S_i)$로 표현한다. 주어진 m비트 TAM 라인의 집합 $TL = \{TL_1, TL_2, \ldots, TL_m\}$라고 하며 TL_i의 길이를 $Length(TL_i)$로 표현한다.

제안된 알고리즘의 부함수로 사용되는 LPT(Largest Processing Time) 함수는 할당 대상인 내부 스캔 체인들을 길이의 내림차순으로 정렬한 후, 각각의 스캔 체인을 현재까지 할당된 스캔 체인의 길이의 총합이 가장 작은 TAM 라인에 할당하는 함수이다. FFD(First Fit Decreasing) 함수는 TAM이 할당받을 수 있는 스캔 체인의 총 길이 합의 한계치 C를 인자로 받아, C를 초과하지 않는 범위 내에서 TAM 라인을 순차적으로 검색하여 내림차순으로 정렬된 각각의 스캔 체인을 할당한다. 만약 한계치 C를 초과할 경우 새로운 TAM 라인에 해당 스캔 체인을 할당한다. LPT 함수와 FFD 함수의 의사코드는 그림 2의 제안된 wrapper 설계 알고리즘 아래 부분에 표현되어 있다.

제안된 wrapper 설계 알고리즘은 우선 내부 스캔 체인의 길이에 대하여 내림차순으로 정렬한다(1번 줄). 내부 스캔 체인을 TAM에 균형있게 할당했을 경우의 이상적인 TAM 라인 길이를 A라고 한다(4번 줄). LPT 수행 결과 리턴 되는 가장 긴 TAM 라인의 길이를 X에 저장한다(5번 줄). X를 FFD 함수의 C의 상한치 CU로 하고(7번 줄) $X/(4/3-1/3m)$와 S_1, A 중에 가장 큰 값을 C의 하한치 C_L로 한다(8번 줄). 한계치 C의 상한치 C_U와 하한치 C_L 사이에 있는 적절한 C값을 찾는다. 이때 C_U에서 값을 감소시키면서 찾음으로써 보다 적은 TAM 너비를 사용하도록 하였다(9-11번 줄). 제안된 코어 테스트 wrapper 설계 알고리즘에서는 LPT 함수를 수행한 후 무조건적으로 FFD 함수를 수행한다. 그리고 두 함수의 결과를 최종 비교하여 wrapper 할당을 결정한다. 이와 같이 FFD 함수와 LPT 함수를 병렬적으로 수행하는 이유는 FFD 함수의 결과보다 LPT 함수의 결과가 더 우수한 경우가 존재하기 때문이고, 또한 두 함수가 모두 동일한 테스트 시간으로 주어진 스캔 체인 셋을 할당할 수 있는 경우 LPT 함수는 항상 FFD 함수보다 같거나 많은 수의 TAM 라인 수를 사용하기 때문이다. 그러므로 LPT 함수는 동일한 테스트 시간을 가지는 FFD 함수의 할당 결과보다 항상 같거나 많은 수의 TAM 라인을 사용한다. 본 논문에서 제안하는 코어 테스트 wrapper 설계 기법은 테스트 시간과 하드웨어 오버헤드를 동시에 고려하면서 Design_wrapper 알고리즘보다 개선된 기법이다.

```
Proposed_wrapper
 1 :  do S descending sort
 2 :  A ← 0
 3 :  for i ← 1 to y
 4 :    do A ← A + Si/m
 5 :  X ← LPT()
 6 :  save the LPT solution
 7 :  CU ← X
 8 :  CL ← ⌈ max(X/(4/3-1/3m), S1, A) ⌉
 9 :  do i ← CU
10 :  while (i ≥ CL) && (FFD(i) < m)
11 :    do i ← i - 1
12 :  if i < CL  then
13 :    do  restore the LPT solution
14 :  end if
15 :  return the solution
```

```
LPT
1 :  do S descending sort
2 :  do TLi = NULL for all j
3 :  for i ← 1 to m
4 :    do TW ← Si
5 :  for i ← m+1 to y
6 :    do select k ∈ {j | Length(TLj) = min1≤x≤m Length(TLx)}
7 :       TLk ← TLk ∪ {Si}
8 :  return max1≤x≤m Length(TLx)
```

```
FFD(C)
 1 :  do S descending sort
 2 :  do TLi = NULL for all j
 3 :  do j ← 1
 4 :  for i ← 1 to y
 5 :    while (1)
 6 :      if Length(TLj) + Si ≤ C
 7 :        then
 8 :        do TLj ← Si;  break;
 9 :      else
10 :        then j ← j + 1
11 :    end if
12 :  return j
```

그림 2. 제안된 wrapper 설계 알고리즘
Fig. 2. Algorithm for Proposed_wrapper.

표 1. 제안된 기법의 할당 결과
Table 1. Proposed_wrapper result for the example.

TAM	Proposed_wrapper
TAM[0]	{12, 3}
TAM[1]	{10, 5}
TAM[2]	{7, 3}

제안된 기법의 간단한 예를 들면, 각각의 길이가 5와 10, 7, 12, 3, 3 인 코어의 내부 스캔 체인이 6개 일 경우, 제안된 알고리즘을 수행한 결과는 표 1과 같다. 이 때 주어진 TAM 라인 개수와 사용한 TAM 라인 개수는 3이고 TAM[0]과 TAM[1], TAM[2] 라인의 길이는 각각 15와 15, 10이므로 이 예제에서의 최장 테스트 시간은 15이다.

할당할 6개의 코어 내부 스캔 체인을 내림차순으로 정렬하면 12, 10, 7, 5, 3, 3이 된다. 제안된 기법은 우선 LPT 함수를 수행한다. LPT 함수 수행 결과, {12, 3}과 {10, 3}, {7, 5}이며 TAM 라인의 길이는 각각 15와 13, 12 로서 최장 TAM 라인의 길이는 15이다. 이 값을 한계치 C의 상한치(C_H)로 사용하여 FFD 함수를 수행하면 각각의 TAM 라인에 {12, 3}과 {10, 5}, {7, 3}을 할당하게 된다. 이 때 두 함수를 수행한 결과를 비교하면 최장 테스트 시간이 15로 같다. 그런데 LPT 함수는 FFD 함수보다 같거나 많은 수의 TAM 라인을 사용하므로 이 경우 FFD 함수의 수행 결과를 취하게 된다. 그러므로 제안된 기법을 수행하여 코어 내부의 스캔 체인을 할당한 결과는 표 1과 같다. 제안된 기법은 알고리즘상의 복잡도는 다소 증가하는 단점이 있지만 하드웨어 오버헤드와 테스트 시간의 감소라는 측면에서 더 효율적이다. 이는 보다 큰 코어에 적용시키면 더 유리한 결과를 보인다.

Ⅳ. 실험 결과

본 논문에서 제안하는 기법의 우수성을 보이기 위해 TAM 너비의 소요 개수와 설계 후 테스트에 필요한 시간을 비교하였다. 실험을 위해서 본 논문에서는 ITC'02 SOC 테스트 벤치마크 회로 중 p93791의 모듈 20 코어를 사용하였다. 모듈 20 코어는 136개의 입력터미널, 12개의 출력터미널, 그리고 72개의 양방향터미널이 있으며, 길이가 132, 133, 157, 168, 180, 181인 내부 스캔 체인이 각각 1, 5, 1, 14, 19, 4개로 총 44개의 내부 스캔 체인을 포함하고 있는 회로이다. 이를 이용한 실험 결과를 표 2에서 보여 준다.

표 2. Design_wrapper와 제안된 기법의 테스트 코어 wrapper 설계 결과
Table 2. Design_wrapper/Proposed_wrapper results for Core 20.

Available TAM width	Utilized TAM width		Longest wrapper scan chain	
	[11]	Prop.	[11]	Prop.
1	1	1	7658	7658
2	2	2	3829	3829
3	3	3	2553	2553
4	4	4	1915	1915
5	5	5	1532	1532
6	6	6	1322	**1309**
7	7	7	1143	**1141**
8	8	8	998	998
9	9	9	865	865
10	10	10	829	829
11	11	11	697	697
12	12	12	685	685
13	13	13	650	**637**
14	14	14	636	**591**
15	15	15	517	517
16	16	16	516	516
17-18	17	17	516	516
19	19	19	481	481
20	20	**19**	480	480
21	21	21	446	**399**
22	22	22	360	360
23-29	23	23	348	348
30	30	30	337	337
31-36	31	31	336	336
37	37	37	325	325
38-39	38	38	301	301
40	40	40	300	300
41-42	41	41	266	266
43	43	43	265	265
44-64	44	44	181	181

표 2의 첫 번째 열은 주어진 TAM 라인 수를 나타낸다. TAM 라인 수가 1비트부터 64비트까지 주어지는 경우를 실험한다. 두 번째([11])와 세 번째 열(Prop.)은 코어의 스캔 체인과 입출력 터미널들을 알고리즘에 따라 할당한 후 최종 사용된 TAM 라인 수를 나타내는데, 두 번째 열은 Design_wrapper 기법[11]의 결과이고 세 번째 열은 제안된 기법의 결과이다. 다섯 번째와 여섯 번째 열은 전체 할당 과정이 완료된 후, 사용된 TAM 라인 중 최장 TAM 라인의 테스트 시간을 클럭

수로 나타낸 것이다. 다섯 번째 열([11])은 Design_wrapper 기법의 결과이고 여섯 번째 열(Prop.)은 제안된 기법의 결과이다.

표 2에서 알 수 있듯이, 주어진 TAM 라인 수가 18인 경우 모든 TAM 라인을 소비하지 않고 17개만을 할당에 소비하면서 주어진 라인 수가 17일 때와 동일한 최장 테스트 시간 결과를 얻을 수 있다. 하지만 표 2의 네 번째 열의 음영처리 된 부분은 본 논문에서 제안하는 기법에 비해 최장 테스트 시간이 더 길다. 제안된 기법은 테스트 시간에 있어서 기존의 기법에 비해 동일하거나 더 나은 결과를 산출한 것을 알 수 있다. 이와 동시에 제안된 기법은 전체적으로 기존의 기법들과 동일하거나 더 적은 TAM 라인을 사용한다. 주어진 TAM 라인이 20개인 부분을 보면 두 번째 열의 굵게 표시된 것처럼 19개의 사용으로 Design_wrapper 기법에 비해 제안된 기법이 더 적은 TAM 라인을 사용한다는 것을 알 수 있다.

V. 결 론

반도체 집적기술이 고도로 발달하면서 출현한 SOC는 미리 설계된 코어를 재사용하는 모듈러 기법을 회로 설계뿐만 아니라 테스트 설계에도 모듈러 기법이 적용되었다. 이러한 SOC 테스트의 비용을 최소화하기 위해서 SOC 테스트 구조의 핵심 구성요소인 코어 테스트 wrapper의 테스트 시간과 면적을 동시에 최적화 시킬 수 있는 설계 기법이 필요하다.

본 논문에서는 SOC 테스트를 위한 비용을 줄일 수 있는 보다 향상된 코어 테스트 wrapper 설계 기법을 제안하였다. 본 논문의 제안 기법은 보다 적은 테스트 시간과 하드웨어 오버헤드를 만족시킨다. 이를 입증하기 위해서 ITC'02 SOC 테스트 벤치마크 회로 중 p93791의 모듈 20 코어를 이용하여 본 논문에서 제안된 방법으로 실험을 하였으며, Intel Pentium-4 1.8GHz CPU, 256MB RAM의 일반적인 PC 시스템 환경에서 C언어를 이용하여 프로그램을 수행하였다. 이러한 실험 결과는 본 논문에서 제안하는 방법의 우수성을 입증하고 있다. 본 논문에서 제안하는 효율적인 코어 테스트 wrapper 설계 기법은 테스트 시간과 하드웨어 오버헤드 비용의 동시 최소화를 만족시키고, 무시해도 좋을 만큼의 시간만을 필요로 하므로, NP-난해 문제인 테스트 코어 wrapper 설계 문제에 실제적으로 적용될 수 있을 것으로 기대된다.

참 고 문 헌

[1] Y. Zorian, E. J. Marinissen, and S. Dey, "Testing Embedded Core-Based System Chips," *IEEE Computer*, Volume 32, Number, 6, pp.52-60, June 1999.

[2] E. J. Marinissen, Y. Zorian, R. Kapur, T. Taylor, and L. Whetsel, "Towards a Standard for Embedded Core Test : An Example," *Proc. International Test Conference*, pp.616-627, 1999.

[3] P1500 Scalable Architecture Task Force Members, "Preliminary Outline of the IEEE P1500 Scalable Architecture for Testing Embedded Cores," *Proc. VLSI Test Symposium*, pp.483-488, 1999.

[4] ITC'02 SOC Test Benchmarks website, "ITC'02 SOC Test Benchmarks", http : //www.extra.research.philips.com/itc02socbenchm/

[5] Peter Harrod, "Testing Reusable IP - A Case Study," *Proc. International Test Conference*, pp.493-498, 1999.

[6] Lee Whetsel, "Addressable Test Ports An Approach to Testing Embedded Cores," *Proc. International Test Conference*, pp.1055-1064, 1999.

[7] Semiconductor Industry Association. Int. Technology Roadmap for Semiconductors, http : // public.itrs.net/files/1999_SIA_Roadmap/Home.htm

[8] Y. Zorian, E. J. Marinissen and S. Dey, "Testing Embedded-Core-Based System Chip,". *Proc. International Test Conference*, pp. 130-143, 1998.

[9] IEEE P1500 General Working Group website, "IEEE P1500 Standards For Embedded Core Test," http : //grouper.ieee.org/groups/1500/

[10] E. J. Marinissen, S. K. Goel, and M. Lousberg, "Wrapper Design for Embedded Core Test," *Proc. International Test Conference*, pp.911-920, 2000.

[11] V. Iyengar. K. Chakrabarty, and E. J. Marinissen, "Test Wrapper and Test Access Mechanism Co-optimization for System On Chip," *Proc. International Test Conference*, pp.1023 -1032, 2001.

[12] R. L. Graham, "Bounds on Multiprocessing Ano-

malies," *SIAM Journal of Applied Mathematics*, Volume 17, pp.416-429, 1969.

[13] E. G. Coffman Jr., M. R. Garey, and D. S. John son, "An Application of Bin-Packing to Multiprocessor Scheduling," *SIAM Journal of Computing*, Volume 7, Number 1, pp.1-17, 1978.

[14] C. Y. Lee, and D. Massey, "Multiprocessor Scheduling : Combining LPT and Multifit," *Discrete Applied Mathematics*, Volume 20, pp.233-242, 1988.

저 자 소 개

최 선 화(정회원)

2002년 단국대학교 전자·컴퓨터공학부 졸업(학사)

2004년 숭실대학교 대학원 컴퓨터학과 졸업(석사)

<주관심분야 : VLSI 설계 및 테스트, 컴퓨터구조, VLSI CAD>

김 문 준(정회원)

2000년 숭실대학교 컴퓨터학부 졸업(학사)

2002년 숭실대학교 대학원 컴퓨터학과 졸업(석사)

2002년~현재 숭실대학교 대학원 컴퓨터학과 박사과정

<주관심분야 : VLSI 설계 및 테스트, 컴퓨터구조, VLSI CAD>

장 훈(정회원)

1987년 서울대학교 전자공학과 졸업(학사)

1989년 서울대학교 전자공학과 졸업(석사)

1993년 University of Texas at Austin 졸업(박사)

1991년 IBM Inc. Senior Member of Technical Staff

1993년 Motorola Inc. Senior Member of Technical Staff

1994년~현재 숭실대학교 컴퓨터학부 부교수

<주관심분야 : VLSI 설계 및 테스트, 컴퓨터구조, VLSI CAD>