

테스트 패턴 재구성을 이용한 NoC(Network-on-Chip)의 저전력 테스트

정준모^{1*}

Low Power Testing in NoC(Network-on-Chip) using test pattern reconfiguration

Jun-Mo Jung^{1*}

요약 본 논문에서는 NoC(Network-on Chip) 구조로 구현된 core-based 시스템에 대한 효율적인 저전력 테스트 방법을 제안한다. NoC의 라우터 채널로 전송되는 테스트 데이터의 전력소모를 줄이기 위해서 스캔 벡터들을 채널 폭만큼의 길이를 갖는 flit으로 분할하고 flit간 천이율(switching rate)이 최소화 되도록 don't care 입력을 할당하였다. ISCAS 89 벤치마크에 대하여 실험을 한 결과, 제안된 방법은 약 35%의 전력 감소를 나타내었다.

Abstract In this paper, we propose the efficient low power test methodology of NoC(Network-on chip) for the test of core-based systems that use this platform. To reduce the power consumption of transferring data through router channel, the scan vectors are partitioned into flits by channel width. The don't cares in unspecified scan vectors are mapped to binary values to minimize the switching rate between flits. Experimental results for full-scanned versions of ISCAS 89 benchmark circuits show that the proposed method leads to about 35% reduction in test power.

Key words : test, NoC(Network-on chip), dft, scan chain, test pattern

1. 서론

여러 개의 IP(Intellectual Property)를 이용하여 시스템을 설계하는 SoC 설계방식은 이동통신 분야 및 멀티미디어 분야의 시스템 설계에 널리 사용되어왔다. 또한 시스템이 대형화됨에 따라 고성능의 프로세서를 포함한 기능화된 IP 코어(Core)들이 많이 내장되었고 코어간의 데이터 교환이 보다 많아지게 되었다. 이러한 특성으로 인하여 기존의 공용버스(shared bus)구조의 SoC 설계 방식으로는 IP 간의 원활한 데이터 통신을 보장할 수 없게 되었고 이에 새로운 설계방식인 NoC(Network-on-chip) 설계방식이 등장하게 되었다[1,2].

NoC 설계방식은 스위칭 네트워크 방식으로 여러 개의 IP를 라우터로 구성하며, 구성된 네트워크를 이용하여 데

이터를 통신하는 방식이다. 이러한 방식은 기가비트(Giga Bit) 수준의 데이터 통신이 가능하고 IP의 재사용성(reusability)과 확장성(scalability)을 가능하게 한다. 따라서 공용버스 구조에 비해서 성능이 훨씬 뛰어나고 전력소모도 작다.

여러 개의 IP 코어가 집적되어 있고 라우터 및 스위치 등의 회로가 내장됨에 따라 NoC를 테스트하기 위해서는 이전보다 많이 복잡해지고 어려워졌다. 테스트 비용으로서 중요한 요소인 테스트 시간을 줄이기 위한 방법들이 제안되었다. 테스트 패턴의 효율적인 스케줄링을 통하여 테스트 시간을 줄이는 방법이 제안되었으며[3], 각 코어의 전력소모를 고려한 테스트 시간 감축방법이 제안되었다[4].

NoC 테스트에 있어서 또 다른 중요한 요소는 테스트 하는 동안에 소모하는 전력이다. 테스트 하는 동안에 과도한 전력소모가 흐르게 되면 칩의 신뢰도가 떨어지고 고장이 유발될 수 있다. 특히 여러 개의 라우터 및 스위치가 내장된 NoC의 경우, 라우터에서 소모하는 전력이 많기 때문에 더욱 중요한 요소가 된다. NoC의 테스트에 관한

이 논문은 2005학년도 군산대학교 신입교수 연구비 지원에 의하여 연구되었음.

¹ 군산대학교 전자정보공학부

*정준모(jmjung@kunsan.ac.kr)

논문은 주로 테스트 시간을 줄이는 스케줄링에 많은 연구가 진행되었지만 저전력 테스트에 대해서는 주목할 만한 연구가 진행되지 못했다.

본 논문에서는 NoC의 테스트 전력소모를 줄이기 위한 효율적인 저전력 테스트 방식을 제안한다. NoC의 채널에서 소모하는 전력을 줄이기 위해서, 각 IP 코어에 입력되는 테스트 패턴을 채널폭만큼의 플릿(flit) 단위로 테스트 패턴을 재구성하고 플릿간 천이가 최소가 되도록 테스트 패턴 내의 don't care 입력을 효율적으로 할당하였다.

2절에서는 NoC의 구조 및 이전의 연구에 대해서 기술하고, 3절은 본 논문에서 제안한 전력소모 모델과 저전력 테스트 패턴 재합성에 대하여 기술하였으며 4절은 실험 결과, 5절은 결론이다.

2. NoC의 구조 및 이전의 연구

NoC는 기존의 SoC내의 IP 코어들을 라우터와 채널로 연결하여 통신하는 구조로 구성된다. 이와 같은 구조를 토폴로지(topology)라고 하며 각 노드에는 라우터/스위치를, 연결선은 채널을 나타낸다. 일반적인 NoC는 grid, torus, mesh, ring 등의 토폴로지로 구성되며 패킷단위로 전송되는 통신용 메시지는 헤더(header), 페이로드(payload) 및 트레일러(trailer)로 구성된다[5].

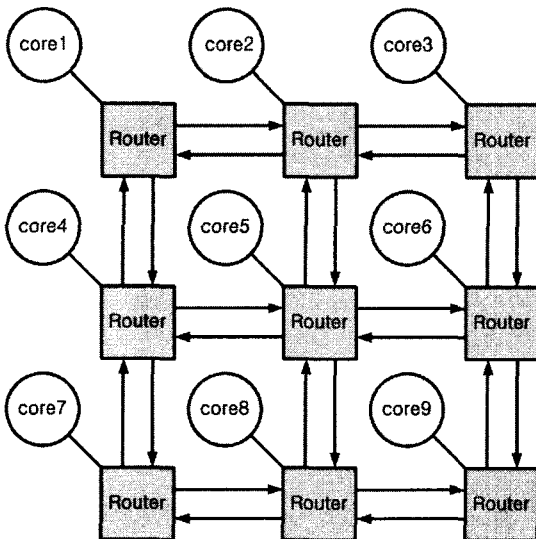


그림 1. 2차원 매쉬(mesh)형 NoC 구조

헤더는 전송되는 패킷의 상태 및 제어신호로 구성되며 페이로드는 전송데이터로 구성된다. 또한 패킷은 플릿

(flit)이라고 하는 작은 규모의 데이터로 분할되어 전송되는데 크기는 NoC 채널의 폭에 맞추어 결정된다. 일반적인 NoC의 채널 폭은 16비트, 32비트 및 64비트로 구성된다. 그림 1은 2차원 매쉬(mesh) 구조로 구성된 NoC를 나타내었다. 기본적인 라우팅 기법은 XY 라우팅이며 스위칭은 웜홀(wormhole)방식이고 각 라우터마다 플릿을 저장하기 위한 버퍼가 내장되어 있으며 버퍼의 오버플로우를 방지하기 위한 플로우 제어회로가 있다.

NoC의 테스트 시간을 줄이기 위한 방법과 기능적 전력소모를 줄이기 위한 연구가 진행되었다. NoC의 테스트 시간을 줄이기 위해서 테스트 스케줄링을 효율적으로 결정하는 방법이 제안되었는데 이 논문에서는 각 코어마다 테스트 시간을 결정 한 후에, 코어를 테스트하는 순서를 적절하게 변경하여 NoC를 테스트하는데 요구되는 시간을 최소화하는 알고리즘을 제안하였다[4]. 또한 각 코어의 전력소모를 고려한 테스트 스케줄링 방법도 제안되었다[6]. 이 논문에서는 각 코어의 테스트 전력소모를 모델링 한 후에, 전력소모를 초과하지 않도록 테스트 시간을 스케줄링하는 알고리즘을 제안하였으나 테스트 전력소모를 줄이지는 못하였다. 최고치(peak) 전력소모를 줄이기 위해서 코어를 효과적으로 배치하는 배치알고리즘을 제안하였다[7]. 전력소모를 최소화하기 위해서 코어간 통신이 많은 것일수록 가깝게 배치하는 등의 휴리스틱한 알고리즘을 제안하였지만 테스트 전력소모는 고려하지 않았다.

3. 전력소모 모델과 저전력 테스트 패턴 재합성

본 절에서는 full scan 회로에 대하여 scan-in 테스트를 하는 동안에 소모되는 scan-in 전력소모 모델 및 스캔 벡터내에 존재하는 don't care 입력을 효과적으로 0 또는 1로 매핑하는 방법을 기술한다.

1) full scan 회로의 테스트 전력소모 모델

먼저 scan-in 전력소모 모델에 대하여 기술한다.

scan-in 테스트의 전력소모는 스캔 벡터가 스캔 체인으로 입력하는 동안 체인에서 발생하는 천이로 인한 동적 전력소모(scan-in 전력소모)와 테스트 응답을 출력하는 동안 체인에서 발생하는 전력소모(scan-out 전력소모)로 나눌 수 있다. 그러나 스캔 벡터로부터 직접 scan-out 전력소모를 계산하는 것이 어렵기 때문에 본 논문에서는 scan-in 전력소모만을 고려하였다.

scan-in 전력소모는 Weighted transitions metric (WTM : 가중 천이 메트릭)모델을 이용하여 측정하였다. 이것은 스캔 벡터의 천이위치를 알면 scan-in 하는 동안 발생하는 천이 수를 측정할 수 있다. 예를 들어 스캔 벡터 $S_1S_2S_3S_4S_5 = 10101$ 이고 스캔 체인 길이가 5인 경우를 고려하자. 가장 왼쪽에 있는 S_1 이 스캔체인에 먼저 입력된다고 가정하면, S_1 과 S_2 에서 발생된 천이(1→0)는 스캔 체인을 통과하면서 총 4번의 천이가 발생한다. 왜냐하면 S_1 과 S_2 에서 발생된 천이는 클럭이 발생할 때마다 (스캔 체인의 크기 - 1) 만큼 이동하기 때문이다. 또한 S_2 와 S_3 에서 발생된 천이(0→1)는 3번의 천이를 발생한다. 이와 같이 S_j , S_{j+1} 에서 발생된 천이는 (스캔 체인의 길이 - j) 만큼의 천이를 발생시킨다.

스캔 체인의 길이를 K라 하면, 각 스캔 벡터 SV는 S_1, S_2, \dots, S_k 로 구성된다.

이때 각 스캔 벡터가 스캔체인에 입력되는 동안 소모되는 전력소모를 WTM으로 표현하면 식(1)과 같이 나타낼 수 있다.

$$WTM(SV) = \sum_{j=1}^{k-1} (S_j \oplus S_{(j+1)}) (k-j) \quad (1)$$

또한 테스트에 이용되는 스캔 벡터의 집합 $SV_{set} = (SV_1, SV_2, \dots, SV_n)$ 이라하면, SV_{set} 이 모두 스캔체인에 입력하는 동안 소모되는 전력소모는 식(2)와 같다.

$$WTM(SV_{set}) = \sum_{i=1}^n WTM(SV_i) \quad (2)$$

식 (2)는 스캔 벡터들을 scan-in 하는 동안 소모되는 전체 전력소모이다. 따라서 평균 전력소모 및 최대치 전력소모는 식 (3)과 같이 나타낼 수 있다.

$$P_{avg} = WTM(SV_{set})/n \quad (3)$$

$$P_{peak} = \text{Maximum}(SV_j)$$

2) 저전력 테스트 패턴 재구성 및 경로 설정방법

NoC 내의 각 코어를 테스트하기 위해서는 코어별 테스트 패턴을 생성해야 하며 생성된 테스트 패턴을 각 코어에 입력해야 하고 테스트 응답을 출력해서 결과를 확인해야 한다. 일반적인 ATPG를 사용하여 테스트 패턴을 생성하면 don't care 입력을 포함하게 된다. 이 don't care 입력은 0 또는 1로 할당되어 코어에 입력되는데 NoC 내의 통신 채널에 맞도록 테스트 패턴을 재구성해야 하며, 저전력으로 don't care를 매핑 하면 테스트 하는 동안의 테스트 전력소모를 줄일 수가 있다. 또한 테스트 대상 코어에 접근하기 위해서는 라우터를 거쳐야 하는

데 전력소모를 최소화하기 위한 접근경로를 찾아야 한다.

① 저전력을 위한 코어별 테스트 패턴의 재구성 방법

코어별 테스트 패턴은 코어의 스캔 체인 길이만큼의 비트 수(k)를 갖는 n개의 테스트 패턴으로 구성된다. 코어에 테스트 패턴을 인가하기 하는 과정은 다음과 같다. 먼저, 테스트 패턴을 채널 폭만큼의 비트 수(w)로 테스트 패턴을 나누어야 한다. k 비트로 구성된 n개의 테스트 패턴은 총 $k*n$ 비트수가 되며 이 비트수를 채널 폭인 w로 나누어 플릿 단위로 전송해야 한다. 플릿 단위로 나뉘어진 테스트 패턴을 테스트 하고자 하는 코어에 인가하기 위해서는 경로를 설정해야 하며 저전력으로 테스트하기 위해서는 가능한 채널 상에 천이가 적게 발생하도록 해야 한다.

우선 플릿 단위로 테스트 패턴을 재구성할 때 적용하는 저전력 방법이다.

저전력 don't care 매핑이다. 일반적으로 don't care가 포함된 테스트 패턴은 1 또는 0으로 할당된다. 본 논문에서 제안하는 방법은 이웃한 플릿 간 데이터의 천이가 적도록 don't care를 할당하는 것이다. 플릿은 라우터의 채널 경로를 통과하여 버퍼에 저장된 후, 코어에 인가된다. 또한 이웃한 플릿들이 연속적으로 채널을 통과하면서 천이가 발생하면 전력이 소모되고 천이가 많을수록 전력소모는 커지게 되며 라우터의 버퍼에서도 동일한 천이가 발생하므로 버퍼의 읽기/쓰기 과정에서도 전력소모가 발생한다. 따라서 이웃한 플릿의 천이율은 전력소모와 밀접한 관계가 있으므로 가능하면 천이가 적게 발생하도록 don't care를 할당하는 것이다. 천이가 적게 발생하기 위해서는 don't care가 있는 플릿의 비트 값을 바로 직전 플릿의 값으로 대체를 한다. 따라서 don't care가 있는 비트위치에서는 천이가 발생하지 않는다.

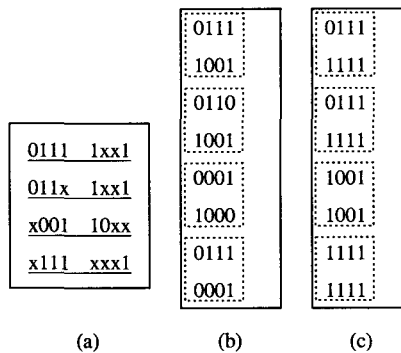


그림 2. 저전력 테스트 패턴 재구성 예

플릿의 스위칭 비율(해밍거리)은 채널에서 소모하는 전력과 각 IP 코어에서 테스트 하는 동안 소모하는 전력에 영향을 주므로 가능한 작도록 해야 한다.

그림 2에 저전력 테스트 패턴 재구성 예를 나타내었다.

스캔 체인의 길이가 8이고 채널 폭이 4라고 가정한다. (a)는 don't care가 포함된 4개의 테스트 패턴이며 (b)는 기존의 0 매핑(don't care에 0을 할당), (c)는 제안된 저전력 매핑이다.

8비트로 구성된 4개의 테스트 패턴을 4비트의 채널로 전송하기 위해서는 한 패턴을 (b)처럼 4비트(플릿단위)로 나누어야 한다.

0 매핑 이므로 모든 don't care 값에 0을 할당하였고 각 플릿의 이웃한 해밍 거리의 합은 20이 된다. 즉 테스트 패턴을 전송하는 동안 채널이나 라우터의 버퍼 등에서 천이하는 스위칭이 20번 발생한다는 것이다. (c)에 나타난 바와 같이, 제안된 방법에서는 don't care 입력 값을 바로 이전의 플릿과 동일한 위치의 값으로 대체하므로 천이가 많이 줄어든다. 저전력 매핑으로 한 경우, 이웃한 플릿의 해밍거리 합은 6이 되어 매우 효율적으로 천이가 감소될 수 있음을 보여준다.

제안한 알고리즘의 Pseudo code를 그림 3에 나타내었다.

```

k: 각 테스트 패턴의 비트 수
n: 테스트 패턴의 개수
w: NoC 채널 폭
test_pattern[i]: 코어에 인가할 i번째 테스트 패턴
flits(i,j): 플릿(i번째 플릿의 j번째 bit)

for(i=0; i<n ; i++) /* 테스트 패턴을 플릿 단위로 분할하여 flits 에 할당*/
    pattern_to_flits(test_pattern[i]);

for(i=1; i<n*k/w ; i++) /* 분할된 테스트 패턴에 don't care 매핑 */
    for(j=0; j < w ; j++)
        if(flits(i,j)=='x') flits(i,j)=flits(i-1,j);
    
```

그림 3. 제안한 알고리즘의 Pseudo Code

② 테스트 경로 설정

테스트 경로는 코어에 테스트 패턴을 인가하기 위해서 생성되어야 한다. 또한 경로를 설정할 때 고려해야 할 것은 패턴의 천이가 많은 것은 짧은 경로를 설정해야 해서 천이가 전파되는 것을 최대한 줄여야 한다. 라우터를 여러 개 거치면 그만큼 천이가 많이 발생하고 전력소모도 커지게 된다. 또한 천이가 적게 발생하는 코어의 테스트 패턴은 경로가 좀 길어도 별 문제가 없으므로 우선순위를 낮게 해도 된다.

효율적인 경로 설정 방법은 다음과 같다.

- 코어별 테스트 플릿에 대하여 천이율을 계산
- 코어별 천이율을 내림차순으로 정리
- 가장 큰 천이율을 갖는 테스트 플릿의 코어를 가장 우선순위가 높도록 하여 접근 경로가 가장 짧도록 설정

4. 실험결과

ISCAS89 벤치마크 회로를 대상으로 제안된 방법을 적용하여 얻은 실험결과에 대해 기술한다. 실험대상회로는 한 개의 스캔체인을 이용하여 full-scan 회로로 구성되었다고 가정한다. MINTEST ATPG(Automatic Test Pattern Generation) 프로그램을 이용하여 dynamic compaction 방법[8]으로 생성된 스캔벡터들을 이용하여 실험하였다. 적용된 NoC 구조는 그림 4와 같으며 6개의 벤치마크 회로(s5378, s9234, s13207, s15850, s38417, s38584)를 코어에 적용하여 실험하였고 ATE 장비로부터 테스트를 입력받아서 적용하는 것으로 설정하였다. 그 외에 두개의 test core를 가정하였다.

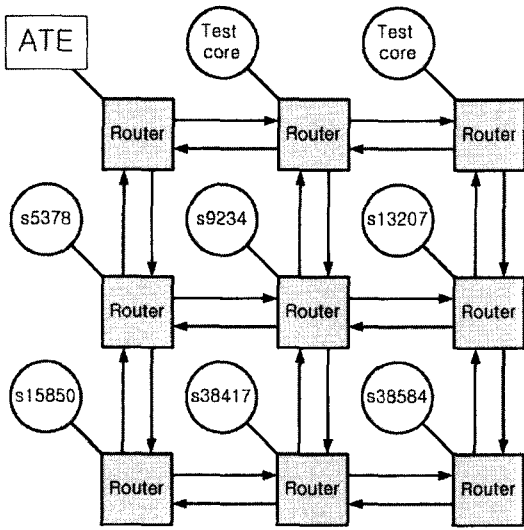


그림 4. 실험환경용 NoC의 구조

표 1 은 각 벤치마크 테스트 패턴에 대하여 제안한 알고리즘을 적용했을 경우, 플릿에서 발생하는 전력소모를 천이의 발생수로 나타낸 것이다. NoC 채널 폭을 16비트, 32비트 및 64비트로 가정하였고 각 채널별로 벤치마크에 대한 전력소모를 계산하였다. 이전의 방법처럼 0 매핑을 한 경우(0 매핑)와 제안한 방법을 적용한 경우, 채널 폭(w)이 16비트인 경우에는 평균적으로 33%의 전력감소율을 보였으며 벤치마크 중에서는 s5378의 전력소모 감소율이 40%가 넘어서 최대로 감소하였다. 또한 w가 32비트인 경우에는 평균 전력소모 감소율이 36%이며 w가 64비트인 경우에는 40%이다. 전력소모 감소율은 채널 폭이 증가함에 따라 증감함을 알 수 있었다.

표 1. 벤치마크별 실험 결과

| 벤치마크 | 전력소모 | | | | | | 전력소모 감소율(%) | | |
|--------|---------------|-------|-------|--------|-------|-------|-------------|------|------|
| | 이전의 방법 (0 매핑) | | | 제안한 방법 | | | | | |
| | w=16 | w=32 | w=64 | w=16 | w=32 | w=64 | w=16 | w=32 | w=64 |
| s5378 | 5546 | 5598 | 5564 | 3240 | 3205 | 3216 | 42 | 43 | 42 |
| s9234 | 7928 | 7970 | 8101 | 5247 | 5271 | 5465 | 34 | 34 | 33 |
| s13207 | 8321 | 8218 | 8382 | 5433 | 5442 | 5423 | 35 | 34 | 35 |
| s15850 | 8370 | 8392 | 8487 | 5662 | 5789 | 5730 | 32 | 31 | 32 |
| s38417 | 39326 | 43981 | 44771 | 24631 | 26682 | 26745 | 37 | 39 | 40 |
| s38584 | 25186 | 24935 | 24977 | 16252 | 16049 | 16073 | 35 | 36 | 36 |
| 평균 | | | | | | | 33 | 36 | 40 |

이러한 코어별 전력소모를 감안하여 NoC에 적용하면 플릿의 천이율이 라우터의 버퍼와 채널에서 그대로 적용이 되므로 전체적인 전력소모는 표 1의 전력 감소 비율과 동일하다고 볼 수 있다. 따라서 본 논문에서 적용한 알고리즘을 이용하여 테스트를 하면 평균 약 30% ~ 40%의 전력소모가 감소된다고 할 수 있다.

5. 결론

NoC의 테스트 전력소모를 줄이는 테스트 패턴 재구성 방법에 대해 제안하였다. 저전력 테스트 패턴을 재구성하기 위해서 플릿으로 분할되는 테스트 패턴내의 don't care 입력 값을 해밍거리가 최소화 하도록 할당하였으며, 코어의 접근경로를 최적화하여 천이가 적게 발생하도록 하였다. ISCAS89 벤치마크 회로에 대해서 실험을 한 결과, 평균 약 35%의 전력소모 감소율을 보였다.

참고문헌

- [1] L. Benini and G. D. Micheli, "Network on Chips : a New SOC Paradigm", IEEE Computer, Jan. 2002.
- [2] C.A. Zeferino, M. Kreutz, L. Carro and A. Susin, "A Study on Communication Issues for Systems-on-Chip", In Symposium on Integrated Circuits and Systems Design, pages 121-126. IEEE Computer Society, Sep. 2002.
- [3] E.cota, M.kreutz, L.carro, M.lubaszweski, A.susin, "The impact of NoC reuse on the testing of core-based systems", Proc. of the 21st IEEE VLSI Test Symposium, Oct. 2003.
- [4] E.cota, L.carro, F.Wagner, M.lubaszweski, "Power-aware noc reuse on the testing of core-based systems," International Test Conference, May, 2003.
- [5] J. Duato et. al. Interconnection Networks : an Engineering Approach. IEEE Computer Society Press, Los Alamitos, CA, 1997.
- [6] C.Liu, V. Iyengar, J. Shi and Erika Cota, "Power-Aware Test Scheduling in Network-on-Chip Using Variable-Rate On-Chip Clocking," Proc. of the 23rd IEEE VLSI Test Symposium, 2005.
- [7] P.Bhojwani, R.Mahapatra, E.J.Kim and T.Chen, "A Heuristic for Peak Power Constrained Design of Network-on-Chip (NoC) based Multimode Systems" Proc. of the 18th International Conference on VLSI

Design, Sep. 2005.

- [8] I.Hamzaoglu and J.H. Patel, "Test set compaction algorithms for combinational circuits", in Proc. Int. Conf. Computer-Aided Design, pp.283-289, Nov. 1998.

정 준 모(Jun-mo Jung)

[종신회원]



- 1987 년 2월 : 한양대학교 전자공학과(공학석사)
- 2004 년 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 전자정보공학부 조교수

<관심분야>

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