

논문 2016-53-8-8

Voltage-Frequency-Island NoC를 위한 테이블 기반의 고장 감내 라우팅 기법

(Table-Based Fault Tolerant Routing Method for
Voltage-Frequency-Island NoC)

윤 성 재*, 이 창 립*, 김 용 석**, 한 태 희***

(Sung Jae Yoon, Chang-Lin Li, Yong Seok Kim, and Tae Hee Han[©])

요 약

반도체 공정 초미세화에 따라 에이징 (aging)과 공정 변이 (process variation)로 인한 칩에서의 물리적인 결함은 더욱 증가하고 있으며, 특히 금속 배선 스케일링 제한과 온 칩 데이터 통신량 증가에 따라 다수의 프로세서 코어로 구성된 네트워크-온-칩(Network-on-Chip, NoC)에서의 결함 감내 기법 연구가 활발히 진행되고 있다. 그러나 NoC에서 저전력 설계 기법으로 널리 채용되고 있는 VFI (Voltage-Frequency-Island)를 적용한 경우에서의 신뢰성 측면에 대한 연구가 부족한 실정이다. 본 논문에서는 신뢰성과 에너지 소모를 고려하여 VFI NoC에서 링크 고장이 발생하는 경우에도 정상적으로 통신을 유지할 수 있는 테이블 기반 라우팅 기법을 제안한다. 에너지 소모와 성능을 고려한 최적 경로와 고장 감내를 위한 우회 경로를 제공하며, 이때 우회 경로의 경우 필요한 최소한의 노드에만 라우팅 테이블을 저장하여 구현 복잡도를 완화하였다. 모의실험 결과를 통해 제안하는 기법은 전체 링크의 1%에서 고장이 발생하는 경우에도 정상적으로 통신함을 보였다. 또한 실시간으로 우회 경로를 탐색하는 고장 감내 라우팅 기법인 d^2 -LBDR에 비해 링크에 고장이 발생하는 경우 평균 15.9%의 에너지 소모가 감소함을 보였으며, 실행 시간 측면에서는 평균 0.8% 감소하는 것을 확인 할 수 있었다.

Abstract

Due to aggressive scaling of device sizes and reduced noise margins, physical defects caused by aging and process variation are continuously increasing. Additionally, with scaling limitation of metal wire and the increasing of communication volume, fault tolerant method in manycore network-on-chip (NoC) has been actively researched. However, there are few researches investigating reliability in NoC with voltage-frequency-island (VFI) regime. In this paper, we propose a table-based routing technique that can communicate, even if link failures occur in the VFI NoC. The output port is alternatively selected between best and the detour routing path in order to improve reliability with minimized hardware cost. Experimental results show that the proposed method achieves full coverage within 1% faulty links. Compared to d^2 -LBDR that also considers a routing method for searching a detour path in real time, the proposed method, on average, produces 0.8% savings in execution time and 15.9% savings in energy consumption.

Keywords : Voltage-Frequency-Island (VFI), Network-on-Chip (NoC), fault tolerant, routing, low power

* 학생회원, ** 정회원, *** 평생회원, 성균관대학교 정보통신대학 (College of Information & Communication Engineering, Sungkyunkwan University)

[©] Corresponding Author (E-mail : than@skku.edu)

※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터 육성지원사업 (IITP-2016- H85 01-16-1005) 및 IT·SW융합원천기술개발 (IITP-2016-R7117-16-0153) 의 연구결과로 수행되었음

Received ; March 16, 2016 Revised ; July 26, 2016

Accepted ; July 29, 2016

I. 서 론

매니코어 (many-core) 프로세서가 보편화 되면서 코어 간의 통신 성능이 전체 칩 성능과 전력소모에 지배적 요소로 대두되고 있으며, 이에 따라 확장성을 고려한 패킷 스위칭 통신 기반의 네트워크-온-칩 (Network-on-Chip, NoC)이 제안되었다^[1]. 또한, NoC 보편화 추세와 인터커넥트에 의한 전력소모 비중이 증가함에 따

라 타일 단위로 특정 전압-주파수 구역화하는 voltage-frequency-island (VFI) 저전력 설계 기법이 많이 적용되고 있다^[2~5].

한편, 최첨단 반도체 칩의 경우 내부 금속 배선 길이의 총합이 수 킬로미터 (km) 수준으로 증가하고 있으며, 또한 공정 미세화로 선폭과 동작 전압이 감소하면서 에이징과 공정 변이로 인해 결함 발생 확률이 기하급수적으로 증가하고 있다^[6]. 이러한 배경에는 EUV (Extreme Ultra-Violet) 노광 장비 개발의 지연으로 인해 다중 패터닝과 광학 근접 보정(Optical Proximity Correction, OPC)에 의한 칩 제조가 이루어지는 것도 주요 요인으로 작용하고 있다^[7~8].

특히 금속 배선의 면적과 전력소모가 칩 전체에서 차지하는 비중이 더욱 커짐에 따라 에이징과 공정 변이로 인해, NoC 구조의 최첨단 시스템-온-칩(System-on-Chip, SoC)에서 배선 고장을 감내할 수 있는 기술의 필요성이 대두되었다. 이에 따라 NoC를 구성하는 링크에서 영구 고장 (permanent fault)이 발생하여 통신이 불가능한 경우 우회 경로를 이용하여 목적지까지 패킷을 전송할 수 있는 다양한 고장 감내 라우팅 기법이 연구되어 왔다^[9~15]. 그러나 에너지 소모 최적화를 위한 VFI 기술이 적용된 NoC에 대해서는 이러한 고장 감내 기법 연구가 부족한 상황이다.

VFI NoC는 특정 어플리케이션에 최적화 된 구조로서 설계 시 실행되는 태스크(task) 특성에 따라 인접한 라우터와 프로세싱 코어들을 VFI 구역으로 묶고 각 VFI 구역마다 서로 다른 전압과 주파수를 할당한다. VFI 구역 간 통신을 위해 각 라우터에 VLC (voltage level converter)와 mcFIFO (mixed clock FIFO)가 필요하며, 패킷이 이를 경유할 때 추가적인 에너지 소모와 지연 시간이 발생한다. 이러한 점들을 고려하여, VFI NoC는 어플리케이션의 트래픽 패턴과 추가 로직을 고려하여 라우팅 경로를 계산하는 것이 보편적이다^[2~5].

하지만 기존 고장 감내 라우팅 기법의 경우 위에서 언급한 요소들을 고려하지 않고, 우회 경로를 계산하여 목적지까지 패킷을 전송하는 것에만 초점을 맞추고 있다. 따라서 VFI NoC에 이를 적용하면 고장 난 링크를 우회하여 통신은 가능하지만 추가 로직을 경유하는 횟수와 패킷이 한 곳에 집중 될 가능성이 증가하여 에너지 소모가 증가하는 문제점이 발생한다.

본 논문에서는 메시 토폴로지 VFI NoC에서의 고장 감내와 더불어 에너지 소모를 최적화할 수 있는 라우팅 테이블 구조를 제안한다. 이를 위해 정상적인 링크 상

태에서의 최적 경로와 링크 고장 발생 시 우회 경로를 제공할 수 있는 라우팅 테이블을 구성하였다. 에너지 소모 측면에서 최적 경로를 계산하기 위해 다익스트라 최단 경로 알고리즘을 사용했으며, 동시에 각 링크의 대역폭 제한 조건을 같이 고려하도록 했다. 또한 우회 경로를 모든 라우팅 테이블에 저장하는 경우 구현 복잡도가 증가하므로 최소한의 라우팅 테이블에 저장을 하는 방식을 이용하여 구현 복잡도를 완화하였다. VFI NoC에 제안하는 기법을 적용하여 링크에서 발생하는 고장 감내 뿐 아니라 통신에서 소모하는 에너지를 최적화하였다.

본 논문의 II장에서는 NoC에서 사용하는 고장 감내 기법에 대한 기존 연구들을 소개한다. III장에서는 제안하는 기법의 필요성, 사용하는 변수, 해결하려는 문제를 기술하고 IV장에서는 신뢰성 향상을 위한 라우팅 할당 기법을 제안한다. V장에서 제안하는 라우팅 기법에 대한 모의실험 결과를 보여주고, 마지막으로 VI장에서 결론을 정리하였다.

II. 링크 고장 감내 NoC 관련 연구

고장 감내 라우팅 기법은 우회 경로를 찾기 위해 사용하는 고장 정보의 범위에 따라 크게 두 가지로 나눌 수 있다. 첫 번째는 NoC를 구성하는 모든 링크의 상태를 탐색하여 우회 경로를 선택하는 기법이다. Fick et al.은 목적지 라우터로 갈 수 있는 최적 경로를, Schonwald et al.은 목적지 라우터까지 전송하는데 필요한 홉 수 (hop count)를 라우팅 테이블에 저장하는 방식을 제안하였다^[9~10]. 그 후, 링크에서 고장이 발생하는 경우 패킷이 경유하는 모든 라우터는 인접 라우터와 고장 난 링크 정보를 교환하면서 우회 경로를 탐색한다. 이 때, 우회 경로가 결정되면 해당 라우터의 라우팅 테이블은 새로운 경로로 갱신된다. 또한 Wachter et al.은 우회 경로를 계산하기 위해 목적지를 찾을 때 까지 인접 라우터로 경로 탐색용 패킷을 순차적으로 전송하는 방식을 제안하였다^[11]. 위에서 소개한 기법은 고장 난 링크의 개수와 상관없이 목적지까지 경로가 존재하는 경우 통신할 수 있는 장점이 있다. 하지만 구현 복잡도가 증가하며 새로운 경로를 계산하고 라우팅 테이블을 재구성하는 동안에는 통신할 수 없는 단점이 있다.

두 번째는 위에서 언급한 단점을 완화하기 위해 인접한 라우터의 링크 상태만 활용하여 우회 경로를 선택하는 기법이다. MD는 신뢰성을 높이기 위해 두 홉 떨어

진 링크 상태를 사용하였고^[12], BLINC는 경로 탐색 속도를 향상하기 위해 출력 포트에 따라 전송 가능한 경로들을 미리 저장하는 방법을 취하였다^[13]. 또한 TOSR은 deadlock을 방지하기 위해 라우터끼리 토큰을 주고받으며 우회 경로를 탐색하거나 라우팅 알고리즘의 복잡도를 줄이기 위해 우회 경로를 선택할 때 최단 경로만을 고려한 연구도 있다^[14-15]. 이러한 기법은 모든 라우터의 링크 상황을 파악할 수 없기 때문에 고장 난 링크의 개수가 증가하는 경우 신뢰성이 감소하는 단점이 있지만 구현이 간단하고 실시간으로 우회 경로를 계산할 수 있는 장점이 있다.

본 논문에서는 설계 시 VFI NoC와 어플리케이션의 특징을 전역적으로 고려하여 에너지 소모를 고려한 최적 경로와 고장 감내를 위한 우회 경로를 계산하는 기법을 제안한다. 또한, 위에서 언급한 고장 감내용 라우팅 테이블의 단점인 갱신 속도를 최소화하기 위하여 최적 경로와 우회 경로를 라우팅 테이블에 동시에 저장하여 실시간성을 보완하였다.

III. VFI NoC 링크 고장으로 인한 문제

1. 기존 고장 감내 라우팅의 문제

VFI NoC는 동일한 라우터를 경유하여도 VFI 구역마다 전압-주파수가 다르게 인가되어 소모하는 에너지가 다르다. 또한 다른 VFI 구역으로 통신하는 과정에서 mcFIFO와 VLC를 경유하기 때문에 추가적인 에너지 소모가 발생한다. 하지만 기존 고장 감내 라우팅 기법은 위와 같은 요소를 고려하지 않고 있다. 그림 1은 전압-주파수 구역을 3개로 나눈 5×5 VFI NoC에서 에너지 소모를 최적화하는 경로와 고장 난 링크를 우회하기 위한 경로를 보여준다. 여기에서 Src와 Dst는 각각 출발지와 목적지 라우터를 의미한다. 그림 1의 (a)는 기존 고장 감내 라우팅 기법을 적용하였고, 그림 1의 (b)는 VFI NoC 특징을 고려하여 우회 경로를 계산하였다. 두 가지 방식은 고장 난 링크를 우회하여 목적지 라우터까지 패킷을 전송할 수 있지만 그림 1의 (a) 기법은 다른 VFI 구역으로 통신하는 횟수도 많으며, (V_3, f_3) 가 (V_2, f_2) 보다 낫다고 가정하면 우회 경로에서 소모하는 에너지가 그림 1의 (b) 기법보다 더 크다. 따라서 VFI NoC의 링크에서 고장이 발생하여 우회 경로를 계산하는 경우, 에너지 소모를 최적화하기 위해서는 목적지까지 갈 수 있는 경로 탐색 이외에 mcFIFO와 VLC를 경유하는

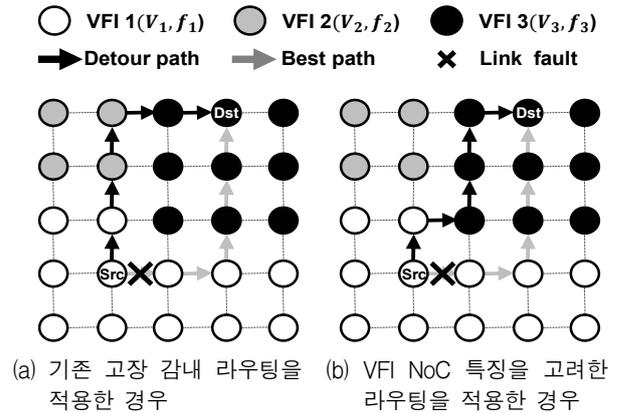


그림 1. VFI NoC에서 링크 고장이 발생 한 예
Fig. 1. An example of link failure in VFI NoC.

횟수를 최소화하고, 각 VFI 구역의 에너지 소모를 고려해야 한다.

2. 문제 정의

제안하는 고장 감내 기법은 최적 경로를 이용해 목적지까지 패킷을 전송하며, 최적 경로가 경유하는 라우터에서 고장을 감지하는 경우 미리 정의되어 있는 우회 경로를 이용해 정상적으로 통신하는 것이 목적이다.

라우팅 경로 계산을 위해 VFI NoC의 mcFIFO와 VLC 뿐 아니라 대역폭을 함께 고려하여 에너지 소모를 최적화할 수 있는 목적함수를 정의하였다. 또한 목적함수를 정의하기 위해 필요한 어플리케이션의 트래픽 패턴과 VFI NoC 특성을 고려한 에너지 모델링을 소개한다.

가. VFI NoC 토폴로지 그래프

VFI NoC에서 라우팅 경로를 효율적으로 할당하기 위해서는 매핑 되어 있는 각 코어의 전압-주파수 특성과 코어 간의 통신 패턴을 고려해야 한다. 논문에서는 이러한 코어 간의 통신을 VFI NoC 토폴로지 그래프 $\mathcal{N}(C, R, S, E)$ 로 정의하였다. 여기에서 $c_i (\in C)$ 는 매핑 되어있는 코어를 의미하고, $r_i (\in R)$ 은 각 코어와 연결된 라우터를 나타낸다. 또한 $s_i (\in S)$ 는 각 코어와 라우터가 어느 VFI 구역에 할당이 되어 있는지를 나타낸다. 또한 $e_{i,j} (\in E)$ 는 c_i 와 c_j 의 통신 관계를 나타내며, $vol(e_{i,j})$ 는 c_i 에서 c_j 로 전송 되는 데이터의 크기를 나타낸다.

나. 에너지 모델링

VFI NoC에서 다른 VFI 구역으로 패킷을 전송하는

경우 발생하는 추가적인 에너지 소모를 다음과 같이 표현할 수 있다^[4].

$$E_{VFI} = E_{CLK} + E_{VLC} + E_{MCFIFO} \quad (1)$$

식 (1)의 E_{CLK} , E_{VLC} , E_{MCFIFO} 는 각각 클럭, VLC, mcFIFO에 의해서 소모되는 에너지를 의미한다. 또한 c_i 에서 c_j 로 전송되는 패킷의 비트 당 에너지를 E_{bit} 로 표현하였다.

$$E_{bit} = \sum_{p \in L(e_{i,j})} E_{L.bit}(p) + \sum_{q \in R(e_{i,j})} (E_{B.bit}(q) + E_{S.bit}(q)) \quad (2)$$

식 (2)에서 $E_{L.bit}$, $E_{B.bit}$, $E_{S.bit}$ 는 각각 VFI NoC의 링크, 버퍼, 스위치에서 소모되는 에너지를 의미하며, $L(e_{i,j})$ 와 $R(e_{i,j})$ 는 경유하는 링크와 라우터가 속하는 VFI 구역을 나타낸다.

다. 에너지 소모를 고려한 목적함수

에너지 소모를 최적화하는 경로를 계산하기 위하여 식 (2)의 에너지 소모 모델링과 다른 VFI 구역으로 통신하는데 소모하는 에너지를 고려하여 목적함수를 아래와 같이 정의하였다.

$$\min \left[vol(e_{i,j})E_{bit} + \sum_{i=1}^m E_{VFI}(i) \right] \quad (3)$$

식 (3)에서 m 은 목적지까지 패킷을 전송하는데 경유하는 VFI 구역의 개수를 나타내며 다음의 대역폭 제한 조건을 만족해야 한다.

$$vol(e_{i,j}) \leq bw_{l,m} \quad (4)$$

식 (4)의 $bw_{l,m}$ 는 r_l 과 r_m 을 연결하는 링크의 실질적으로 사용 가능한 최대 대역폭을 나타낸다. 이러한 제약 조건을 적용하여 패킷 간 병목현상을 최소화하였다.

IV. 고장 감내를 위한 VFI NoC 라우팅 알고리즘

이번 장에서는 링크에서 발생하는 고장을 감내하기 위한 라우팅 테이블 구조를 소개하고, 여기에 저장되는 최적 경로와 우회 경로를 계산하는 알고리즘을 제안한다. 또한 제안하는 라우팅 기법은 deadlock 방지와 처리량 향상을 위해 각 라우터의 입력 포트에 가상 채널을 사용하였다^[21].

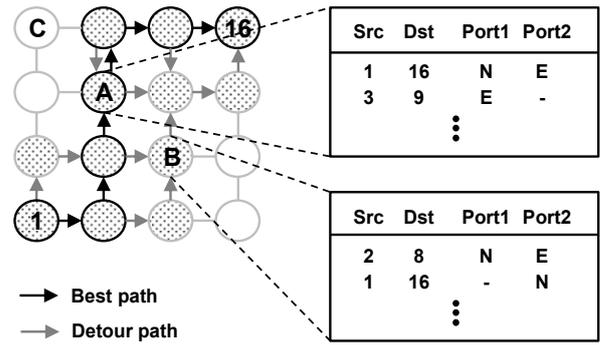


그림 2. 라우터 역할에 따른 라우팅 테이블 구조
Fig. 2. Routing table structure according to the router role.

1. 라우팅 테이블 구조

제안하는 라우팅 테이블은 VFI NoC의 각 라우터마다 연결되어 있는 구조이며, 입력 포트에 들어 온 패킷의 출력 포트를 결정하는 라우팅 계산 과정에서 사용된다. 이를 통해 라우팅 계산 과정이 생략되며, 미리 저장된 최적 경로를 통해 패킷을 전송하지 못할 경우 우회 경로를 통해서 목적지까지 패킷을 전송할 수 있다. 고장 감지의 경우 유사한 다른 연구에서와 같이 각 라우터에서 실시간으로 출력 포트의 고장을 감지하는 것으로 가정하였다^[20].

그림 2는 다음 절에서 소개하는 라우팅 알고리즘을 적용해 테이블을 초기화한 상태를 나타낸다. Src는 시작 라우터를 의미하며 Dst는 목적지 라우터를 의미한다. 또한 링크에 고장이 발생하지 않은 경우 포트 1에 저장되어 있는 경로를 통해서 다음 라우터로 전송을 하고, 링크에 고장이 발생하면 전송 중인 패킷은 포트 2에 저장되어 있는 경로를 사용해서 목적지까지 전송한다. 여기에서 N, E, W, S는 각각 북쪽, 동쪽, 서쪽, 남쪽 방향을 의미한다.

예를 들어 r_1 에서 r_{16} 으로 가는 최적 경로가 경유하는 r_A 는 포트 1과 2에 각각의 방향이 저장되어 있고, 우회 경로가 경유하는 r_B 의 경우는 포트 2에만 방향이 저장되어 있다. 하지만 r_C 의 경우는 라우팅 경로가 경유하지 않기 때문에 r_1 과 r_{16} 에 대한 정보가 저장되어 있지 않다. 이를 통해 우회 경로를 계산하는 과정과 라우팅 테이블을 갱신하는 과정이 생략되며, 미리 저장되어 있는 최적 경로와 우회 경로를 통하여 고장을 감내한다.

2. 라우팅 경로 선택

그림 3의 의사코드는 IV.1절에서 소개한 라우팅 테이블에 저장되는 라우팅 경로를 계산하는 알고리즘을 나

타낸다. 제안하는 라우팅 기법은 대역폭 제한 조건을 만족시키기 위해 *Sorting()*을 이용해 통신 관계가 존재하는 모든 코어 간 전송되는 데이터의 크기를 기준으로 내림차순 정렬한다 (Line 1).

Algorithm : Proposed Routing Path Allocation

```

Input :  $N(C, R, S, E)$ 
1:  $Comm[X] = Sorting(vol(e_{i,j}))$ 
2:  $Update\_cost(N(C, R, S, E))$ 
3: for  $k=0$  to  $X-1$  do
4:    $start = Comm[k].start$ 
5:    $end = Comm[k].end$ 
6:    $tmp\_link = Remove\_link(Comm[k].bw)$ 
7:    $Best\_path = Dijkstra(start, end)$ 
8:    $Update\_routing\_table(Best\_path)$ 
9:    $Update\_link\_bw(Best\_path)$ 
10:  for  $l = Hop\_count$  of  $Best\_path$  to 0 do
11:     $disconnecting\_link(Best\_path[l])$ 
12:     $tmp\_paths = Find\_detour\_path(Best\_path[l])$ 
13:    if there is overlapped detour path
14:       $Detour\_path = overlapped\ tmp\_paths$ 
15:    else
16:       $Detour\_path = compare\_energy(tmp\_paths)$ 
17:    for  $q=0$  to  $Hop\_count$  of  $Detour\_path$  do
18:      if selected  $Routing\_table$  is not empty then
19:        break
20:      else if selected router is destination router then
21:        break
22:      else
23:         $Update\_routing\_table(Detour\_path)$ 
24:         $Update\_link\_bw(Detour\_path)$ 
25:      end for
26:       $connecting\_link(Best\_path[l])$ 
27:    end for
28:     $Recover\_link(tmp\_link)$ 
29:  end for
    
```

Output : *Routing Table of Each Router*

그림 3. 제안하는 라우팅 알고리즘의 의사코드
Fig. 3. Pseudo-code of proposed routing algorithm.

가. 링크의 가중치를 이용한 라우팅 경로 선택

최적 경로는 링크에 고장이 발생하지 않은 상황에서 에너지 소모를 최소화하면서 네트워크의 통신성능 요구를 만족시킬 수 있는 경로를 의미한다. 기존 NoC에서는 홉 수가 동일한 라우팅 경로는 트래픽을 무시하는 경우, 어떠한 경로로 패킷을 전송해도 전체적인 에너지 소모나 데이터 처리량에 차이가 없다. 하지만 VFI NoC는 다른 VFI 구역으로 통신하는 경우 VLC와 mcFIFO에서 추가적인 에너지 소모가 발생하기 때문에 최단 경

로뿐만 아니라 통신이 가능한 모든 경로를 비교해서 최적의 라우팅 경로를 계산해야 한다.

최적 경로는 식 (3)에서 소개한 목적함수를 최적화하는 경로를 계산하기 위하여 구현이 간단한 다익스트라 최단 경로 알고리즘을 사용하였다. 왜냐하면 VFI NoC는 설계 시 라우팅 경로를 계산하므로 재탐색 과정이 없기 때문이다. 알고리즘 구현 과정에서 정점 (vertex)은 라우터와 코어 쌍을 의미하며, 패킷 전송 과정에서 발생하는 에너지 소모를 가중치로 사용한다. 가중치의 경우 r_i 와 r_j 를 연결하는 링크에 식 (2)의 $E_{bit}(r_i)$ 를 할당하였고, 다른 VFI 구역을 연결하는 링크는 식 (1)의 E_{VFI} 를 추가적으로 할당한다 (Line 2). 또한 VFI NoC의 데이터 처리량과 알고리즘 속도 향상을 위하여 식 (4)의 대역폭 제한 조건을 만족하지 못하면 해당 링크를 알고리즘 과정에서 제거한다 (Line 6). 그 후, *Dijkstra()*를 이용하여 최적 경로를 계산하고, 선택된 경로가 공유하는 라우터의 라우팅 테이블과 링크의 대역폭을 갱신한다 (Line 7-9). 이를 통해 최단 경로뿐만 아니라 목적지로 갈 수 있는 모든 경로 중에 에너지 소모를 최소화하는 경로를 선택한다.

나. 고장 감내를 위한 우회 라우팅 경로 선택

그림 4는 최적 경로가 각 라우터에 할당되어 있는 상황에서 우회 경로를 계산하는 알고리즘의 각 단계를 보여준다. 우회 경로를 위한 라우팅 테이블의 개수를 최

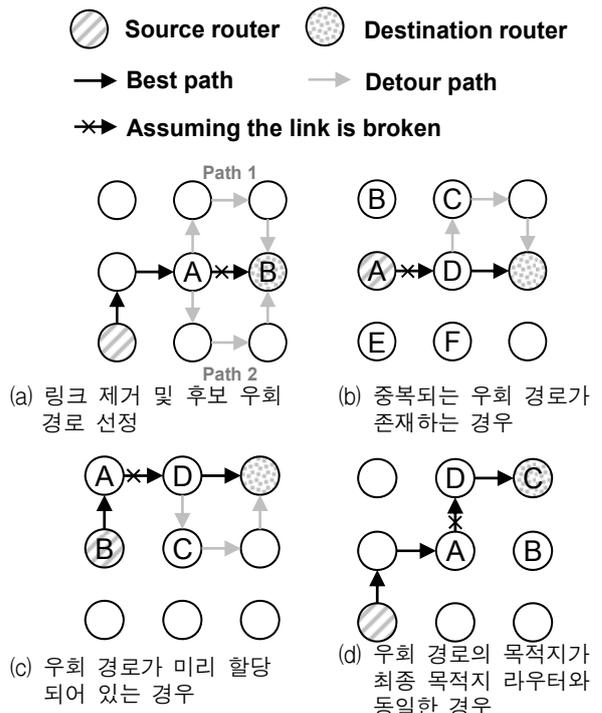


그림 4. 고장 감내를 위한 우회 경로 계산 알고리즘
Fig. 4. Computing the detour path to tolerate link faults.

소화하기 위하여 최종 목적지 라우터와 인접한 링크부터 역순으로 제거한다 (Line 11). 그림 4의 (a)에서는 r_A 와 r_B 를 연결하는 링크가 없음을 가정하였고, $Find_detour_path()$ 를 이용해 r_A 와 r_B 를 연결하는 경로를 탐색한다 (Line 12). 이 때, 그림 4의 (b)와 같이 중복 되는 우회 경로가 존재하면 우회 경로의 개수를 최소화하기 위하여 $r_A \rightarrow r_B \rightarrow r_C \rightarrow r_D$ 를 우회 경로로 선택한다 (Line 13-14). 한 개 혹은 두 개의 후보 우회 경로가 선택되면 $compare_energy()$ 로 에너지 소모가 적은 우회 경로를 선택한다 (Line 16). 우회 경로가 공유하는 라우터 4개가 선택되면 해당 라우터의 라우팅 테이블 갱신 여부를 판단한다. 그림 4의 (c)에서는 후보 우회 경로가 $r_A \rightarrow r_B \rightarrow r_C \rightarrow r_D$ 순으로 선택되는데 r_C 와 같이 동일한 목적지에 해당하는 우회 경로가 미리 존재하거나 그림 4의 (d)와 같이 공유하는 라우터가 최종 목적지와 동일한 경우 해당 라우팅 테이블을 갱신하지 않는다 (Line 18-21). 위와 같은 과정을 통해 어플리케이션을 구동하는데 필요한 모든 통신에 대하여 최적 경로와 우회 경로를 각 라우터에 할당하였다.

V. 비교 및 분석

VFI NoC 특징을 고려한 최적 경로와 우회 경로에서 전체적인 실행 시간과 에너지 소모를 측정하기 위해 SystemC 기반의 사이클 단위 정밀도 시뮬레이터 환경을 구축하였으며, Orion 2.0을 사용하여 소모하는 에너지를 모델링하였다^[16]. 실험용 벤치마크는 MCSL에서 제안한 것을 사용했으며, 어플리케이션을 구성하는 각 태스크는 통신량에 따라 메시 토폴로지 NoC에 할당되어 있다^[17]. 표 1은 실험에서 벤치마크로 사용한 어플리케이션의 특징을 나타낸다.

제안하는 기법의 성능을 비교하기 위하여 낮은 전압-주파수가 할당 된 VFI 구역으로 통신하여 에너지 소모를 최적화한 라우팅 기법 (Opt_{energy})과 높은 전압-주파수가 할당 된 VFI 구역으로 통신하여 지연시간을 최적화한 라우팅 기법 ($Opt_{latency}$)을 적용했다. 또한 링크 고장의 발생 유무에 따른 성능 비교를 위하여 기존 NoC에서 경량 라우터 구조를 적용한 고장 감내 라우팅 기법인 d^2 -LBDR을 사용했다^[15].

각 라우팅 알고리즘의 공정한 성능 평가를 위해 동일한 VFI NoC 환경을 구현한 후, 8×8 과 16×16 메시 토폴로지서 각각 실험을 진행했다. VFI 구역마다 최적의

표 1. 어플리케이션 벤치마크 정보^[17]
Table1. Information of benchmark^[17].

Application (benchmark)	Description	No. tasks	No. edges
H264-720p_dec (Bench. 1)	H.264 video decoder with a resolution of 720p	2311	3461
H264-1080p_dec (Bench. 2)	H.264 video decoder with a resolution of 1080p	5191	7781
FFT-1024_complex (Bench.3)	Fast Fourier Transform with 1024 complex number inputs	16384	25600

전압-주파수를 인가하기 위해^[2]에서 소개한 기법을 적용하여 4개의 VFI 구역으로 나누었으며, 각 구역의 전압/주파수는 (0.6V, 0.78GHz), (0.7V, 1.27GHz), (0.8V, 1.81GHz), (0.9V, 2.42GHz)으로 할당하였다. 또한 가상 채널을 적용한 라우터는^[18]에서 제안한 구조를 사용했다. 이는 5개의 포트에 구성되어 있으며, 각 포트는 4개의 버퍼로 이루어진 4개의 가상채널을 포함한다. 실험에서 고장 난 링크의 비율은 기존 고장 감내 라우팅 연구의 실험을 참조하였으며, 이를 통하여 전체 링크의 1% 이하가 고장 난 경우를 고려하였다^[12~15].

1. 고장 감내율

8×8 메시 토폴로지를 구성하는 전체 링크는 224개이며, 1%에 해당하는 링크는 대략 두 개다. VFI NoC의 메시 토폴로지는 모서리 부근에 고장 난 링크의 개수가 두 개인 경우에는 경로가 존재하지 않으므로 어떠한 라우팅 기법을 적용하여도 통신을 할 수 없다^[19]. 실험에서는 위와 같은 상황을 배제하였으며, 그 결과 제안하는 기법은 임의로 선택한 링크에 고장이 발생하는 경우에 우회 경로를 통하여 고장을 100% 확률로 감내함을 보였다.

2. 에너지 소모

그림 5는 라우팅 기법에 따른 에너지 소모를 비교한 결과를 나타내며, 고장 난 링크가 없는 상황에서 제안하는 기법에서 소모한 에너지를 기준으로 정규화 했다.

고장 난 링크가 없을 때 제안하는 기법은 $Opt_{latency}$ 에 비하여 에너지 소모가 평균 4.3% 감소함을 보였다. 하지만 라우팅 테이블 크기의 증가와 우회 경로를 선택하는 로직의 추가로 인하여 Opt_{energy} 에 비해서는 평균 5.5%의 에너지 소모가 증가하는 결과를 보였다.

d^2 -LBDR과 비교하면 링크 고장이 없는 상황에서 평균 10.5%의 에너지 감소를 보였으며, 고장이 발생하

는 경우 평균 15.6%의 에너지 감소를 보였다. 어플리케이션에 따라서는 H264-720p_dec와 트래픽 패턴이 비교적 복잡한 FFT-1024_complex에서 각각 8.1%, 13.2%의 에너지 소모가 감소하는 것을 보였다. 이를 통하여 제안하는 라우팅 방식이 VLC와 mcFIFO를 경유하는 홉 수를 최소화하였고, 각 링크의 대역폭을 고려하여 통신 효율성이 증가하면서 고장 감내 뿐만 아니라 에너지 소모도 최적화한 것으로 해석할 수 있다.

또한 네트워크 크기와 트래픽 패턴에 따른 에너지 소모를 비교해 보았을 때, 제안하는 라우팅 기법은 네트워크 크기가 증가하고 트래픽 패턴이 복잡할수록 더욱 효율적인 것으로 볼 수 있다. 특히 16×16 토폴로지에 FFT-1024_complex를 할당한 환경에서 제안하는 기법

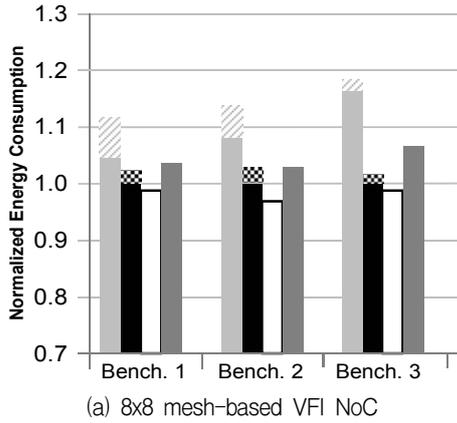
은 d^2 -LBDR에 비하여 에너지 소모가 24.7% 감소하는 것을 보였다.

3. 어플리케이션 실행 시간

그림 6은 고장 난 링크가 없는 상황에서 제안하는 라우팅 기법에서 소모한 실행 시간을 기준으로 정규화한 결과를 보여준다.

고장 난 링크가 없는 상황에서 Opt_{energy} 의 경우 낮은 전압-주파수 VFI 구역으로 통신하는 빈도의 증가로 인하여 제안하는 기법과 비교해 평균 3.4%의 실행 시간이 증가하는 것을 보였다. $Opt_{latency}$ 는 높은 전압-주파수 VFI 구역으로 통신하여 실행 시간을 단축시키지만 특정 구역에 패킷이 집중되면서 실행 시간이 평균

■ d^2 -LBDR (No link fault) ▨ d^2 -LBDR (1% link fault)
 ■ Proposed (No link fault) ▩ Proposed (1% link fault)
 □ Opt_{energy} (No link fault) ■ $Opt_{latency}$ (No link fault)



■ d^2 -LBDR (No link fault) ▨ d^2 -LBDR (1% link fault)
 ■ Proposed (No link fault) ▩ Proposed (1% link fault)
 □ Opt_{energy} (No link fault) ■ $Opt_{latency}$ (No link fault)

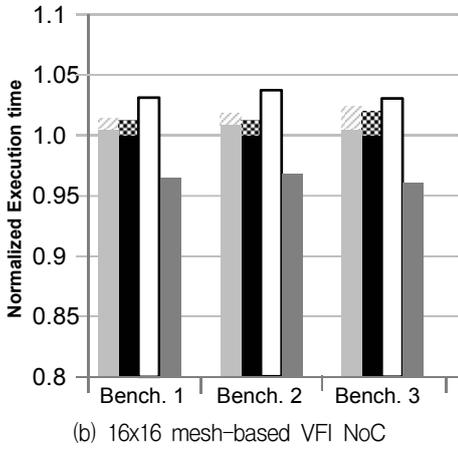
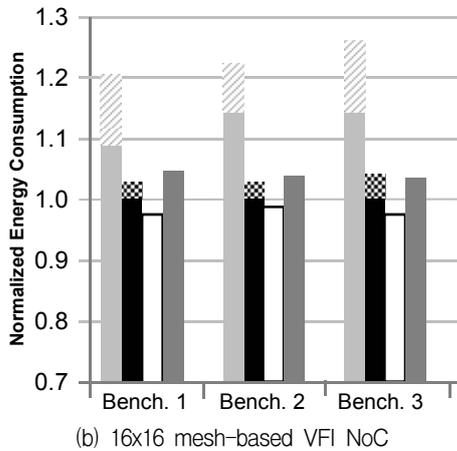
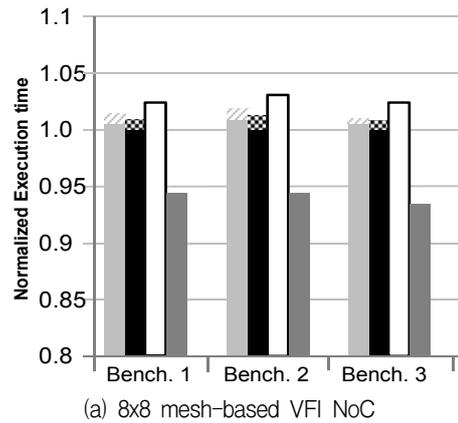


그림 5. 어플리케이션과 네트워크 크기에 따른 에너지 소모량 비교
 Fig. 5. Comparison of the energy consumption under different application and network size.

그림 6. 어플리케이션과 네트워크 크기에 따른 실행시간 비교
 Fig. 6. Comparison of the execution time under different application and network size.

4.1% 감소하는데 그쳤다.

제안하는 기법은 에너지 소모를 최소화하는 경로를 선택하기 때문에 실행 시간이 늘어나는 경로가 존재한다. 하지만 최적 경로와 우회 경로를 라우팅 테이블에 미리 저장하여 새로운 경로를 계산하는데 소모하는 시간을 최소화하였기 때문에 실시간으로 경로를 계산하는 d^2 -LBDR과 비교하였을 때 실행 시간 측면에서 평균적으로 0.8%의 미세한 감소를 보였다.

이를 통하여 제안하는 기법은 링크에서 고장이 발생하는 경우 미리 저장되어 있는 우회 경로를 통해 정상적으로 통신을 유지할 수 있으며, 기존 고장 감내 라우팅 기법과 비교하여 실행 시간은 유지하면서 에너지 소모는 감소하는 것을 보였다.

4. 고장 감내 유무에 따른 면적

라우터 면적 측면에서 제안하는 기법의 오버헤드를 측정하기 위하여 Verilog-HDL로 설계한 라우터를 Cadence의 RTL compiler와 Nangate 45nm cell 라이브러리를 이용하여 합성한 후 면적을 측정하였다^[22].

표 2는 라우터를 구성하는 각 모듈의 면적을 보여준다. baseline은 최적 경로만을 저장하였으며, proposed는 최적 경로와 우회 경로를 모두 저장하였다. 또한 제안하는 기법은 라우터마다 라우팅 테이블의 개수가 다르므로 전체 라우팅 테이블 개수의 평균값을 기준으로 측정하였다.

실험 결과를 보면 트래픽 패턴이 복잡할수록 추가 되는 우회 경로가 많아져 면적이 증가함을 보였다. 하지

표 2. 단일 라우터의 각 모듈별 면적 크기 및 우회 경로 추가에 따른 면적 오버헤드

Table2. Area of each module in a single router and area overhead due to the additional detour path.

Module		Baseline	Proposed
FIFO		25,940 μm^2	
VC Allocator		6,472 μm^2	
Switch Allocator		1,830 μm^2	
Crossbar		3,783 μm^2	
Routing Computation	Bench. 1	1,180 μm^2	1,420 μm^2
	Bench. 2	1,315 μm^2	1,515 μm^2
	Bench. 3	1,770 μm^2	2,135 μm^2
Overhead	Bench. 1	0.59 %	
	Bench. 2	0.49 %	
	Bench. 3	0.87 %	

만 라우터를 구성하는 모듈 중 FIFO의 경우 63% 이상의 면적을 차지하는 반면에 경로 계산 모듈은 5% 미만을 차지한다. 결과적으로 고장 감내를 위해 라우팅 테이블에 우회 경로를 추가해도 라우터의 면적은 1% 미만으로 증가한 것을 볼 수 있다.

VI. 결 론

본 논문에서는 VFI NoC 특징을 고려하여 고장 감내 이외에 에너지 소모를 최적화하는 라우팅 테이블 기법을 제안하였다. 고장 난 링크를 우회하는 경로를 찾는 것 이외에 VFI NoC 특징을 고려하여 경로를 탐색하였다. 또한 최적 경로와 우회 경로를 동시에 라우팅 테이블에 저장하여 고장 감내를 위한 경로 탐색과 라우팅 테이블 갱신 과정을 생략하였다.

제안하는 기법의 성능을 평가하기 위하여 SystemC 기반의 시뮬레이터를 작성하고, 고장 감내 라우팅 기법인 d^2 -LBDR을 적용하여 에너지 소모와 실행 시간을 비교하였다. 그 결과 VFI NoC를 구성하는 링크에서 고장이 발생하여도 목적지까지 통신을 할 수 있었으며, 어플리케이션과 네트워크 크기에 상관없이 실행 시간은 평균적으로 0.8% 감소하였으며, 에너지 소모는 트래픽 패턴과 네트워크 크기에 따라 최대 24.7% 감소했다. 또한 제안하는 기법으로 인하여 경로 계산 모듈의 면적이 증가하지만 라우터의 면적 증가는 1% 미만임을 보였다.

REFERENCES

- [1] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," Computer, vol. 35, no. 1, pp. 70-78, Jan. 2002.
- [2] W. Y. Jang, D. Ding, and D. Z. Pan, "A Voltage-Frequency Island Aware Energy Optimization Framework for Networks-on-Chip," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 1, no. 3, pp. 420-432, Sept. 2011.
- [3] C.-L. Li, J. H. Lee, J.-S. Yang, and T. H. Han, "Communication-aware custom topology generation for VFI network-on-chip," IEICE Electronics Express, vol. 11, no. 18, pp. 1-8, Sept. 2014.
- [4] U. Ogras, R. Marculescu, D. Marculescu, and E. Jung, "Design and Management of Voltage-Frequency Island Partitioned Networks-on-Chip," IEEE Trans. Very Large Scale Integration System, vol. 17, no. 3, pp. 330-341, March 2009.

- [5] S. Herbert, S. Garg, and D. Marculescu, "Exploiting Process Variability in Voltage/Frequency Control," *IEEE Trans. Very Large Scale Integration Systems*, vol. 20, no. 8, pp. 1392-1404, Aug. 2009.
- [6] A. DeOrio, D. Fick, V. Bertacco, and D. Sylvester, "A Reliable Routing Architecture and Algorithm for NoCs," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 5, pp. 726-739, May 2012.
- [7] J. D. Owens, W. J. Dally, R. Ho, D. N. Jayasimha, S. W. Keckler, and L. S. Peh, "Research Challenges for On-Chip Interconnection Networks," *IEEE Micro*, vol. 27, no. 5, pp. 96-108, Sept.-Oct. 2007.
- [8] L. E. Bechtold, D. Redman, and B. Tawfello, "Semiconductor Reliability Using Random and Wearout Failure Models," in *Proc. of 2014 Annual Reliability and Maintainability Symp. (RAMS)*, pp. 1-6, Colorado Springs, CO, USA, Jan. 2014.
- [9] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw, "A Highly Resilient Routing Algorithm for Fault-Tolerant NoCs," in *Proc. of Design, Automation and Test in Europe Conf. and Exhibition (DATE)*, pp. 21-26, Nice, France, April 2009.
- [10] T. Schonwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel, "Fully Adaptive Fault-Tolerant Routing Algorithm for Network-on-Chip Architectures," in *Proc. of 10th Euromicro Conf. Digital System Design Architectures, Methods and Tools (DSD)*, pp. 527-534, Lubeck, Germany, Aug. 2007.
- [11] E. Wachter, A. Erichsen, A. Amory, and F. Moraes, "Topology-Agnostic Fault-Tolerant NoC Routing Method," in *Proc. of Design, Automation and Test in Europe Conf. and Exhibition (DATE)*, pp. 1595-1600, Grenoble, France, March 2013.
- [12] M. Ebrahimi, M. Daneshtalab, J. Plosila, and F. Mehdipour, "MD: Minimal path-based Fault-Tolerant Routing in On-Chip Networks," in *Proc. of 18th Asia and South Pacific Design Automation Conf. (ASP-DAC)*, pp. 35-40, Yokohama, Japan, Jan. 2013.
- [13] D. Lee, R. Parikh, and V. Bertacco, "Brisk and Limited-Impact NoC Routing Reconfiguration," in *Proc. of Design, Automation and Test in Europe Conf. and Exhibition (DATE)*, pp. 1-6, Dresden, Germany, March 2014.
- [14] M. Balboni, J. Flich, and D. Bertozzi, "Synergistic Use of Multiple On-Chip Networks for Ultra-Low Latency and Scalable Distributed Routing Reconfiguration," in *Proc. of Design, Automation and Test in Europe Conf. and Exhibition (DATE)*, pp. 806-811, Grenoble, France, March 2015.
- [15] R. Bishnoi, V. Laxmi, M. S. Gaur, and J. Flich, "d2-LBDR: Distance-Driven Routing to Handle Permanent Failures in 2D Mesh NoCs," in *Proc. of Design, Automation and Test in Europe Conf. and Exhibition (DATE)*, pp. 800-805, Grenoble, France, March 2015.
- [16] A. B. Kahng, B. Li, L. S. Peh, and K. Samadi, "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration," in *Proc. of Design, Automation and Test in Europe Conf. and Exhibition (DATE)*, pp. 423-428, Nice, France, April 2009.
- [17] W. Liu, J. Xu, X. Wu, Y. Ye, X. Wang, W. Zhang, M. Nikdast, and Z. Wang, "A NoC Traffic Suite Based on Real Applications," in *Proc. of IEEE Computer Society Annual Symp. VLSI (ISVLSI)*, pp. 66-71, Chennai, India, July 2011.
- [18] Robert Mullins, Andrew West, and Simon Moore, "Low-Latency Virtual-Channel Routers for On-Chip Networks," in *Proc. of the 31st annual International Symposium on Computer Architecture (ISCA)*, pp. 188-197, May 2004.
- [19] Y. C. Chang, C. T. Chiu, S. Y. Lin, and C. K. Liu, "On the Design and Analysis of Fault Tolerant NoC Architecture Using Spare Routers," in *Proc. of 16th Asia and South Pacific Design Automation Conf. (ASP-DAC)*, pp. 431-436, Yokohama, Japan, Jan. 2011.
- [20] A. Prodromou, A. Panteli, C. Nicopoulos, and Y. Sazeides, "NoCAlert: An On-Line and Real-Time Fault Detection Mechanism for Network-on-Chip Architectures," in *Proc. of 45th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 60-71, Vancouver, Canada, Dec. 2012.
- [21] Y. J. Yoon, N. Concer, M. Petracca, and L. Carloni, "Virtual Channels vs. Multiple Physical Networks: a Comparative Analysis," in *Design Automation Conf. (DAC)*, pp. 162-165, Anaheim, USA, June 2010.
- [22] "The Nangate Open Cell Library, 45nm FreePDK", available online at <https://projects.si2.org/openeda.si2.org/projects/nangatelib/>.

저 자 소 개



윤 성 재(학생회원)
2015년 서경대학교 전자공학과 학사 졸업.
2015년 3월~현재 성균관대학교 정보통신대학 석사과정.
<주관심분야: SoC 설계, NoC>



이 창 림(학생회원)
2010년 연변과학기술대학교 컴퓨터 전자통신공학부 학사 졸업.
2012년 부산대학교 인지메카트로닉스공학과 석사 졸업.
2012년 9월~현재 성균관대학교 정보통신대학 석박사과정.
<주관심분야: SoC 설계 및 테스트>



김 용 석(정회원)
1983년 성균관대 전자공학과 졸업.
1983년 1월~1997년 12월 삼성 전자 DM연구소 ASIC설계 그룹장.
1998년 1월~2009년 12월 삼성전자 통신연구소 IMT2000모뎀 SW그룹장 및 삼성전자 음향기술센터장.
1999년~2002년 IDEC KAIST(반도체설계교육센터) 운영위원.
2004년 1월~2013년 12월 삼성전자 상무.
2010년 1월~2013년 12월 삼성전자 무선사업부 시스템SW팀장. (GalaxyS1,2,3/Note1,2 사업화)
2014년 3월~현재 성균관대 산학협력단 및 정보통신대학 교수, 기업지원거점센터장, IoT 사업화 지원센터장.
<주관심분야: SoC 아키텍처 설계, 시스템소프트웨어, IoT 시스템설계>



한 태 희(평생회원)
1992년 KAIST 전기 및 전자공학과 학사 졸업.
1994년 KAIST 전기 및 전자공학과 석사 졸업.
1999년 KAIST 전기 및 전자공학과 박사 졸업.
1999년 3월~2006년 8월 삼성전자 통신연구소 책임 연구원.
2006년 9월~2008년 2월 한국산업기술대학교 전자공학과 조교수.
2008년 3월~현재 성균관대학교 정보통신대학 반도체시스템공학과 부교수.
2011년 5월~2013년 4월 지식경제부 시스템반도체 PD.
<주관심분야: SoC 아키텍처 및 설계 방법론, 3D IC, 메모리/스토리지 시스템 구조, 임베디드 SW, IT 융합기술>