

논문 2016-53-4-8

# 버스 프로토콜 호환 가능한 네트워크-온-칩에서의 분리된 주소/데이터 네트워크 설계

( Separated Address/Data Network Design for Bus Protocol compatible Network-on-Chip )

정 승 아\*, 이 재 훈\*, 김 상 현\*, 이 재 성\*, 한 태 희\*\*

( Seungh Ah Chung, Jae Hoon Lee, Sang Heon Kim, Jae Sung Lee, and Tae Hee Han<sup>©</sup> )

## 요 약

다중 프로세서 시스템-온-칩(Multi-Processor SoC, MPSoC)에서의 코어 및 IP 개수 증가 추세에 따라 병렬처리와 확장성에 유리한 인터커넥션 구조인 네트워크-온-칩(Network-on-Chip, NoC)이 등장하였다. 하지만 기존 IP를 재활용하기 위해서는 버스 프로토콜과 호환가능한 NoC에서의 지연시간을 최소화하기 위한 연구가 필요하다. 본 논문에서는 버스 프로토콜 호환 가능한 NoC 설계 시, 버스 프로토콜에서 특성이 다른 다수의 트랜잭션 단계에서 유발되는 홉 수와 경로 충돌의 대립관계로 인해 지연시간이 증가하는 문제를 주소 및 데이터 네트워크로 분리 설계함으로써 해결하였다. 모의실험으로 벤치마크 어플리케이션과 무작위 생성한 어플리케이션에서의 실험 결과를 통해 Mesh구조와 TopGen의 비정형 토폴로지와 비교했을 때, 평균 지연시간은 19.46% 및 실행시간은 10.55% 감소하였다.

## Abstract

As the number of cores and IPs increase in multiprocessor system-on-chip (MPSoC), network-on-chip (NoC) has emerged as a promising novel interconnection architecture for its parallelism and scalability. However, minimization of the latency in NoC with legacy bus IPs must be addressed. In this paper, we focus on the latency minimization problem in NoC which accommodates legacy bus protocol based IPs considering the trade-offs between hop counts and path collisions. To resolve this problem, we propose separated address/data network for independent address and data phases of bus protocol. Compared to Mesh and irregular topologies generated by TopGen, experimental results show that average latency and execution time are reduced by 19.46% and 10.55%, respectively.

**Keywords :** NoC, Topology generation, Latency, Contention, Bus protocol

## I. 서 론

다중 프로세서-온-칩(Multi-Processor SoC, MPSoC)에서의 코어 및 IP 개수 증가 추세에 따라, 온-칩 인터커넥트 구조 설계가 중요해지고 있다. 전통적인 버스 구조에서는 연결되는 컴포넌트 수가 증가함에 따른 병목현상으로 인해 확장성에 한계가 있다<sup>[1]</sup>. 이러한 문제를 해결하기 위해서 병렬처리와 확장성에 유리한 네트워크-온-칩 (Network-on-Chip, NoC)이 유망한 인터커넥션 아키텍처 기술로 주목 받고 있다.

하지만 현재 표준화된 NoC 프로토콜의 부재 및 기존 버스 프로토콜로 설계되어진 legacy IP의 호환성 문제<sup>[2]</sup>

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

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

※ 본 논문은 산업통상자원부의 “웨어러블 디바이스를 위한 경량 운영체제 기반 초저전력 대기모드 핵심기술 개발” (10054668) 과제 및 2015년도 정부(교육부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(NRF-2015R1D1A1A01057278)

Received ; February 2, 2016 Revised ; March 28, 2016

Accepted ; April 4, 2016

등으로 인해 NoC의 성능을 제대로 활용하지 못하고 있다. 특히 확장성에 초점을 맞춘 NoC 설계는 지연시간에 크리티컬한 디스플레이 IP 및 실시간 처리가 필요한 어플리케이션에서는 설계의 검증 및 데드라인을 충족하기 위한 최적화를 필요로 한다.

NoC에서의 지연시간은 데이터가 경과하는 홉 수와 직접적인 연관이 있는데, 이를 최적화하기 위해 이분 그래프 (Bipartite graph)를 이용한 라우터 할당 기법 및 MCF (Multicommodity Flow) 기반의 경로 계산을 적용한 토폴로지 합성 방법<sup>[3]</sup>, 링크 추가를 통한 대체 경로 생성 방법, 메시지 스케줄링을 적용하여 지연시간을 개선하는 방법<sup>[4]</sup> 등이 제시되었다. 이와 같이 지연시간 개선을 위한 기존 연구에서 경로 및 링크의 특성은 반영되었으나, 버스 프로토콜 호환 가능한 NoC에서 버스 프로토콜의 트랜잭션 지연시간 최적화를 위한 연구는 이뤄지지 않았다.

하나의 버스 프로토콜 통신은 주소 및 데이터, 응답이라는 다단계 (Multiple Phase) 트랜잭션 (Transaction)이 수행되어야 완료된다. 따라서 기존 버스 프로토콜 기반 IP를 재사용하기 위한 NoC 설계 시, 다단계 버스 트랜잭션의 지연시간에 영향을 주는 홉 수의 최적화가 고려되어야 한다. 하지만, 홉 수를 줄이기 위한 설계 기법은 경로의 다양성을 저해하여 경로 충돌 확률을 증가시킨다<sup>[5]</sup>. 경로 다양성의 보장은 중재 시간(Arbitration time) 및 라우터 복잡도를 고려할 때, 제한적인 라우터의 포트 수에서 경로의 홉 수를 증가시킨다. 따라서, 이러한 대립 관계를 고려한 설계 방식의 연구가 필요하다.

본 논문에서는 기존 버스 프로토콜 기반 IP의 재사용 가능한 NoC 설계에 있어, 버스 프로토콜의 주소와 데이터 전송 단계를 고려하여 주소/데이터 네트워크를 분리한 토폴로지 설계 기법을 제안한다. 분리된 네트워크는 주소와 제어 정보를 전송하는 주소 네트워크에서는 홉 수의 최소화, 데이터 전송량이 상대적으로 훨씬 많은 데이터 네트워크에서는 경로 충돌 문제를 해결하기 위해서 경로 다양성을 보장하는 방향으로 토폴로지를 생성한다.

본 논문의 II장은 NoC 설계에서 다중 네트워크를 사용하는 연구들에 대하여 기술하며, III장에서 분리된 주소/데이터 네트워크를 위한 토폴로지에서 최적화 설계 문제에 대하여 정의하고, IV장에서는 제안하는 주소 네트워크와 데이터 네트워크 토폴로지 생성 알고리즘 및 데이터 네트워크 경로 제어 설정을 기술한다. V장에서는 제안하는 토폴로지 생성 알고리즘에 대한 모의실험 결과를 보

여주고, 마지막으로 VI장에서 결론을 정리한다.

## II. 관련 연구

최근 들어 이종 매니코어 (Heterogeneous manycore)로 구성된 NoC 설계 시 다수의 네트워크를 사용하여 에너지 효율 또는 성능을 높이는 연구가 등장하고 있다. Balfour et al.은 CMesh와 같은 정형 토폴로지에 병렬적인 서브 네트워크 (sub-network)을 도입하여 트래픽을 분산시킴으로써 성능을 높이는 이종 멀티 네트워크를 제안하였고<sup>[6]</sup>, Mishra et al.은 대역폭과 지연시간 각각에 최적화된 분리 네트워크를 사용하는 NoC 설계 방식을 제안하였다<sup>[7]</sup>. 또한 A. Lusala에서는 서킷 스위칭 (Circuit Switching) 방식과 패킷 스위칭 (Packet Switching) 방식을 모두 사용할 수 있는 하이브리드 라우터를 공간 분할 다중화 (Spatial Division Multiplexing) 설계 방식에 적용하여 서킷 스위칭 네트워크에서는 스트리밍 트래픽을, 패킷 스위칭 네트워크에서는 최선형 (Best-Effort) 트래픽을 전송함으로써 QoS (Quality-of-Service) 관점에서 성능을 개선하였다<sup>[8]</sup>.

요약하자면, 기존의 다중 네트워크 연구들은 NoC에서의 어플리케이션 특성 또는 스위칭 방식을 고려한 성능 개선에 집중되었다. 하지만 기존 버스 프로토콜로 설계된 IP의 재사용이 중요시됨에 따라, 버스 프로토콜 호환 가능한 NoC 설계에서의 성능을 높이기 위해서 다단계로 구성된 버스 프로토콜의 트랜잭션에 최적화된 다중 네트워크 구조에 대한 연구가 필요하다.

## III. 주소/데이터 네트워크 설계 문제 정의

본 장에서는 버스 프로토콜 호환 가능한 NoC 설계 시 고려해야 할 사항을 살펴보고, 이를 위한 최적화 설계 문제를 정의하고자 하며 이때 고려해야 할 사항은 다음과 같다. 첫째로, 버스 프로토콜 상 발생하는 연결형 (connection-oriented) 전송 특성은 다단계로 구성된 트랜잭션으로, 많은 홉 수의 통신에서 비례적으로 지연시간이 증가하므로 이를 최소화해야 한다. 둘째로, 병렬성을 최대화하는 동시에 경로충돌을 최소화해야 한다.

위의 설계 방식에 대해 버스 프로토콜 관점에서 설계 문제를 정의하면 다음과 같다.

1) 주소 및 응답 단계는 비교적 적은 양의 정보 전송을 수행하며, 실질적인 데이터 전송을 위한 통신 준비 단계로, 경로 충돌보다 홉 수가 지연시간에 더 큰 영향

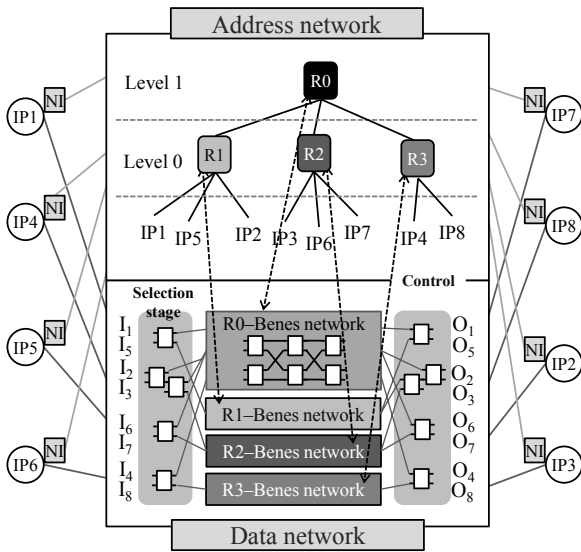


그림 1. 제안한 주소/데이터 네트워크  
Fig. 1. Proposed Address/Data network.

을 미친다.

2) 데이터 단계는 대량 전송이 이루어지므로 장시간 경로 점유로 인하여 경로 충돌 확률이 증가한다. 따라서, 경로 충돌이 지연시간에 더 큰 영향을 미친다.

위의 1)과 2)의 문제를 모두 만족시키기 위해서는 적은 홉 수로 경로 충돌을 최소화시키는 네트워크를 설계해야 한다. 하지만, 두 가지 목적은 대립적인 관계를 가지므로 이에 대한 trade-offs를 고려해 지연시간 최적화를 달성해야 한다<sup>[5]</sup>.

이런 제약 조건을 고려하여, 본 논문에서는 그림 1과 같이 버스 프로토콜 호환 가능한 NoC에서 주소 네트워크와 데이터 네트워크를 분리 설계하여 이를 해결하고자 한다. 서로 다른 목적을 갖는 분리된 네트워크 설계를 위해 다음과 같은 정의를 기반으로 연구를 진행하였다.

가. 커뮤니케이션 추적 그래프

코어 간의 통신을 나타내는 커뮤니케이션 추적 그래프 (Communication Trace Graph, CTG)는  $G(V, E)$ 로 표현되며,  $v_i \in V$ 는 코어를 나타내고  $e_{i,j} \in E$ 는 코어  $v_i$ 가  $v_j$ 로 데이터를 전송하는 통신을 나타낸다. 또한 각  $e_{i,j}$ 는 토폴로지를 설계하기 위한 두 가지 정보를 갖는다: 1)  $BW(e_{i,j})$ 는  $v_i$ 가  $v_j$ 로 보내는 통신의 대역폭을 나타낸다. 2)  $CV(e_{i,j})$ 는 데이터를 전송할 때 한 번의 통신에서 보낼 수 있는 최대 데이터양을 나타낸다.

나. 이종 (Heterogenous) 네트워크 토폴로지

토폴로지 설계의 결과는 두 가지 네트워크 토폴로지 그래프로 분류되며, 이는 다음과 같다.

1) 주소 네트워크 토폴로지 그래프는  $AT(U, R, F)$ 로 표현되며,  $u_i \in U$ 는 토폴로지에서의 코어 및 IP를 나타내고  $r_i \in R$ 는 토폴로지에 존재하는 각각의 라우터를 나타낸다.  $f_{u_i, r_j} \in F$ 는 코어와 라우터 사이의 연결인  $(u_i, r_j)$ 를 나타내고  $f_{r_i, r_j} \in F$ 는 라우터와 라우터 사이의 연결인  $(r_i, r_j)$ 를 나타낸다.

2) 데이터 네트워크 토폴로지 그래프는  $DT(U, C, F)$ 로 표현되며,  $u_i \in U$ 는 토폴로지에서의 코어 및 IP를 나타내고  $c_i \in C$ 는 토폴로지에 존재하는 각각의 크로스바를 나타낸다.  $f_{u_i, c_j} \in F$ 는 코어와 크로스바 사이의 연결인  $(u_i, c_j)$ 를 나타내고  $f_{c_i, c_j} \in F$ 는 크로스바와 크로스바 사이의 연결인  $(c_i, c_j)$ 를 나타낸다.

다. 이종 토폴로지 설계 문제

본 논문에서 제안하는 이종 토폴로지 설계 문제는 다음과 같이 정의된 토폴로지 설계 함수  $TG()$ 를 찾는 문제이다.

$$TG(G(V, E)) = \{AT(U, R, F), DT(U, C, F)\}$$

이 때, 토폴로지 설계 함수  $TG()$ 는 다음과 같은 세 가지 조건을 만족하여야 한다.

1) 주소 네트워크에서의 지연시간은 주로 홉 수의 영향을 받으므로 식 (1)와 같이 표현되는 홉 수의 총합을 최소화시켜야 한다.

$$Com_{total} = \sum_{\forall e_{i,j}} \frac{BW(e_{i,j})}{CV(e_{i,j})} \times hops(e_{i,j}) \quad (1)$$

2) 데이터 네트워크에서는 경로 충돌을 최소화하기 위해, 각 링크에 중복되는 통신의 수를 최소화시키는 네트워크 토폴로지를 생성해야 한다.

3) 주소 정보를 주소 네트워크가 가지고 있기 때문에 데이터 네트워크의 경로 설정을 주소 네트워크에서 제어해야한다. 따라서, 주소 네트워크의 라우터들이 데이터 네트워크를 분산 제어할 수 있어야 한다.

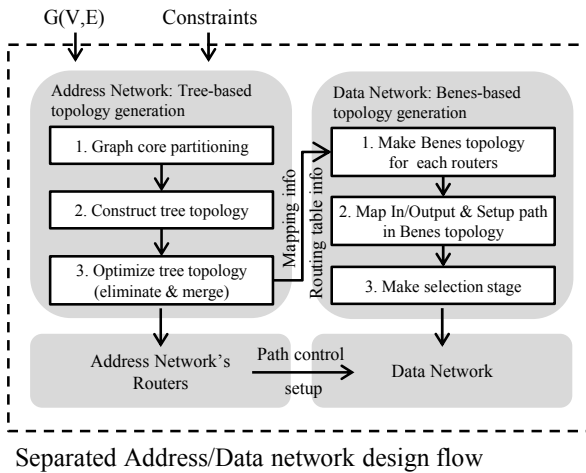


그림 2. 제안하는 분리된 네트워크 설계 흐름도  
Fig. 2. Flow of proposed separated network design.

#### IV. 제안하는 주소/데이터 토폴로지 설계 기법

통상적으로, 토폴로지 설계 기법은 NP-hard 문제로 가능한 모든 토폴로지를 고려한 솔루션 도출이 힘들다. 따라서, 본 논문에서는 기존 정형 토폴로지 기반에서의 설계 및 최적화 기법을 결합한 이중 토폴로지 설계 기법을 제안한다. 제안하는 토폴로지 설계는 그림 2과 같은 흐름으로 진행된다. 1) 주소 네트워크를 위한 트리 구조 기반의 토폴로지 생성, 2) 데이터 네트워크를 위한 베니스(Benes) 구조 기반의 토폴로지 생성 및 데이터 네트워크의 경로 제어를 위한 설정.

##### 1. 주소 네트워크 토폴로지 생성 알고리즘

트리 구조 기반의 토폴로지는 홉 수를 최적화함과 동시에 확장 가능한 솔루션을 제공할 수 있는 정형적인 토폴로지이다<sup>[9-10]</sup>. 우선, 트리 토폴로지는 계층적으로 구성되는데 통신이 상위 라우터로 한 레벨씩 지나갈 때 마다 홉 수가 증가되기 때문에, 상위 레벨을 지나는 통신을 줄이는 것이 전체적인 홉 수 감소에 효과적이다. 트리 토폴로지의 홉 수는 다음 식 (2)으로 도출될 수 있다.

$$hop(e_{i,j}) = 2(LH_{e_{i,j}} - LL_{e_{i,j}}) + 1 \quad (2)$$

여기서  $LH_{e_{i,j}}$ 는  $e_{i,j}$ 가 경과하는 상위 라우터의 레벨,  $LL_{e_{i,j}}$ 는 하위 라우터의 레벨을 나타낸다. 그림 3과 같이 CTG를 다수의 그래프로 분할하여 트리 토폴로지로 매핑했을 때, 서로 다른 서브-그래프로 분할되는 통

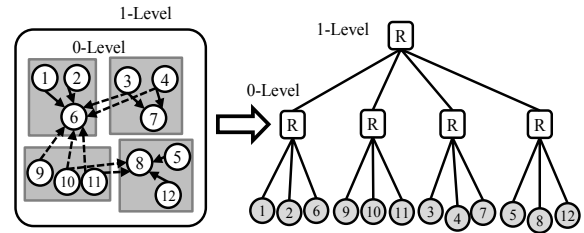


그림 3. 그래프 분할을 통한 트리 토폴로지 생성  
Fig. 3. Tree topology generation by graph partitioning.

##### Algorithm 1 Address part - Tree-based topology generation

**Input:**  $n$ : number of node,  $p$ : number of router's port  
 $G(V, E)$ : CTG  
**Output:**  $AT(U, R, F)$ : generated tree topology  
1: **partition**  $G(V, E)$  to  $p$  sub-graph minimized  $Com_{total}$   
2: **foreach** sub-graph **do**  
3: **while**  $Num_{node} > p - 1$   
    //  $Num_{node}$ : number of node included in current sub-graph  
4: **partition** sub-graph to  $p - 1$  sub-graph minimized  $Com_{total}$   
    // up to here, generate symmetry & regular tree topology  
5: **remove** the routers where no node connected  
6: **foreach** level  $l - 1$  router **do** //  $l$ : last level of routers  
7: **if**  $node_{curr} \leq p$  **then** //  $node_{curr}$ : current connected total nodes on descendant routers  
8: **merge** the descendant routers with number of  $node_{curr}$   
    // up to here, eliminate redundancy and merge routers  
9: **return**  $AT(U, R, F)$

그림 4. 주소 네트워크를 위한 트리 구조 기반의 토폴로지 생성 알고리즘  
Fig. 4. Tree-based topology generation algorithm for address network.

신은 상위 라우터를 지나게 되어 홉 수가 증가한다. 따라서 주소 네트워크 설계는 서로 다른 서브-그래프와의 통신 수를 최소화시키는 균등 그래프 분할(Balanced graph partitioning) 문제<sup>[11]</sup>로 귀결될 수 있다.

균등 그래프 분할 기반의 주소 네트워크 설계 알고리즘은 그림 4의 의사 코드(Pseudo code)로 표현된다.

노드들 간의 전체 통신량을 최소화하는 트리 토폴로지를 생성하기 위해서, 우선적으로 전체 노드 수  $n$  과 라우터의 포트 수  $p$ 에 따라  $Com_{total}$ 을 최소화하는 그래프 분할을 진행한다. 최상위 라우터는 상위 라우터로의 링크가 없으므로, 최상위 라우터에 연결 가능한 서브-그래프는  $G(V, E)$ 를  $(p, 1 + p/n)$ -그래프 분할을 통하여 생성된다 (Line 1). 반면 최상위 라우터를 제외한 하위 라우터에서는 각 서브-그래프별로  $(p - 1, 1 + (p - 1)/n)$ -그래프 분할을 수행한다. 이 때, 하나의 서브-그래프에 배치되는 최대 노드의 수는 다음

과 같은 수식 (3)을 따르게 된다.

$$\max |V_i| = (1 + p/n) \left\lceil \frac{|V|}{p} \right\rceil \quad (3)$$

즉, 배치되는 최대 노드의 수는 분할하고자 하는 그래프에 포함된 코어의 수를  $p$ 로 나눈 숫자보다 하나의 노드를 더 가질 수 있다. 따라서 그래프 분할을 각 서브-그래프에 포함되는 코어의 수가  $p-1$  이하가 될 때까지 수행하며, 결국 리프 라우터에는  $p-1$  이하의 코어가 연결된 트리 구조 기반 정형 토폴로지가 구성된다 (Line 2-4).

그래프가 불균등하게 분할되었을 때 서브-그래프에 노드가 존재하지 않거나 라우터 포트 수 보다 적은 노드 수를 갖는 토폴로지가 생성될 수 있다.

위의 상황은 결국 많은 홉 수를 유발시키는 요인이 될 수 있기 때문에 다음의 추가적인 최적화 단계를 거치게 된다. 1) 노드가 연결되지 않은 라우터의 제거 (Line 5), 2)  $(l-1)$ 레벨 라우터를 기준으로 통합이 가능한 리프 라우터들이 있다면 하나의 리프 라우터로 통합 (Line 6-8)

## 2. 데이터 네트워크 토폴로지 생성 알고리즘

데이터 네트워크는 경로 충돌을 최소화시키기 위해 경로의 다양성을 보장하여야 한다. 또한, 그림 1에서와 같이 주소 네트워크에서 데이터 전송 경로를 설정할 때, 분산 제어가 가능한 데이터 네트워크 설계를 필요로 한다. 따라서, 경로의 다양성을 보장할 수 있는 베니스 토폴로지<sup>[12]</sup> 기반의 데이터 네트워크 설계 기법을 제안한다.

데이터 네트워크는 주소 네트워크의 라우터를 통해 데이터 전송 경로가 제어되도록 설계해야 한다. 이를 위해, 주소 네트워크의 계층적인 트리 토폴로지 구조를 고려하여 다음과 같은 방식으로 데이터 네트워크를 구성하고 경로를 제어한다. 1) 데이터 네트워크는 주소 네트워크의 각 라우터가 제어할 수 있는 종속된 네트워크들로 구성된다. 2) 데이터 네트워크의 입력과 출력에서 각 통신이 지나가야 되는 종속 네트워크를 선택하는 단계를 생성한다. 3) 데이터 전송 경로는 해당 통신이 지나가는 주소 네트워크의 라우터들에 의해 제어된다. 그 중 가장 상위 라우터에서는 해당 종속 네트워크를 제어하며, 나머지 라우터에서는 선택 단계를 제어하여 데이터를 종속 네트워크까지 전달한다.

그림 5는 데이터 네트워크 토폴로지 설계 흐름도를

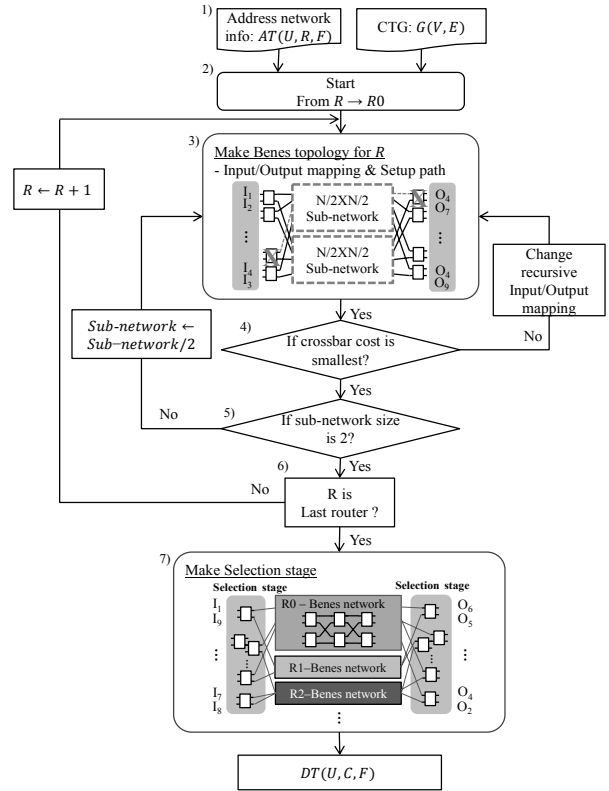


그림 5. 데이터 네트워크를 위한 베니스 구조 기반의 토폴로지 생성 흐름도

Fig. 5. Benes-based topology generation flowchart for Data network.

나타낸다.

1) 데이터 네트워크는 주소 네트워크 정보와 CTG를 기반으로 생성된다.

2) 데이터 네트워크는 주소 네트워크의 각 라우터마다 제어할 수 있는 베니스 기반의 종속 네트워크로 구성되며 최상위 라우터에서부터 순차적으로 생성한다.

3) 각 종속 네트워크의 설계 방식은 다음과 같다. 입력과 출력 수  $N$ 개를 가지는  $N \times N$  베니스 토폴로지는 입력, 출력 단계와 2개의  $N/2 \times N/2$  서브-네트워크 (Sub-network)로 구성된다<sup>[13]</sup>. 경로 설정 방식은 입력과 출력단에서 2개의 서브-네트워크 중 하나를 선택을 통한 정적인 경로를 설정한다.

4) 경로 설정 시 네트워크의 입력과 출력 매핑에 따라 경로 설정이 달라지고, 이에 따라 불필요한 크로스바 스위치를 제거할 수 있다. 따라서 크로스바 스위치를 최소화하는 입력과 출력 매핑을 선택한다.

5)  $N \times N$  네트워크에서의 입력, 출력 매핑과 경로 설정이 끝나면, 내부의  $N/2 \times N/2$  서브-네트워크도 3), 4)를 적용한다. 서브-네트워크의 크기가  $2 \times 2$  될

표 1. 각 커뮤니케이션 추적 그래프의 정보  
Table 1. Information of CTG.

Benchmark	Number of nodes	Number of edges
MPEG4	12	26
DVOPD	32	44
N64	64	126
N100	100	181

때까지 입출력 매핑과 경로 설정을 진행하며, 결과적으로 통신을 고려하여 스위치 개수를 최적화한  $N \times N$  베니스 토폴로지가 생성된다.

6) 이와 같은 방식으로 주소 네트워크의 각 라우터를 위한 베니스 기반의 종속 네트워크를 모두 생성한다.

7) 마지막으로 데이터 네트워크의 노드 입력과 출력에서 각 라우터 마다 생성된 종속된 네트워크들로 갈 수 있는 선택단계를 생성한다.

## V. 실험

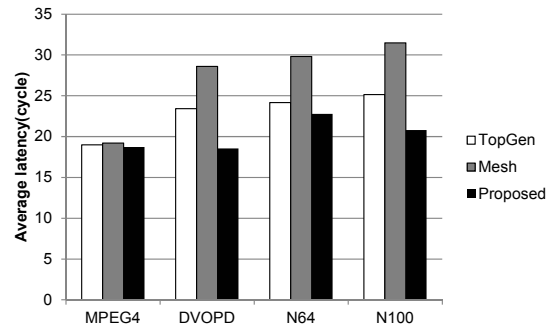
본 논문에서는 분리된 주소/데이터 네트워크의 검증 을 위해 SystemC 기반의 사이클 단위 정밀도 시뮬레이 터를 구축하였다.

이러한 환경 하에서 분리된 네트워크 성능 비교를 위 해 벤치마크 어플리케이션인 DVOPD, MPEG4<sup>[14]</sup>와 TGFF<sup>[15]</sup>로 2개의 커뮤니케이션 추적 그래프를 생성하 여 사용하였다. 커뮤니케이션 추적 그래프들 상에서 제 안하는 버스 호환 가능한 분리된 주소/데이터 네트워크 의 성능을 하나의 네트워크로 구성된 NoC와 비교하기 위해서 다음의 다른 두 가지 구조를 적용한 NoC 네트 워크와 비교 하였다: 1) 정형 토폴로지인 Mesh구조<sup>[16]</sup> 2) 비정형 토폴로지의 TopGen<sup>[17]</sup>.

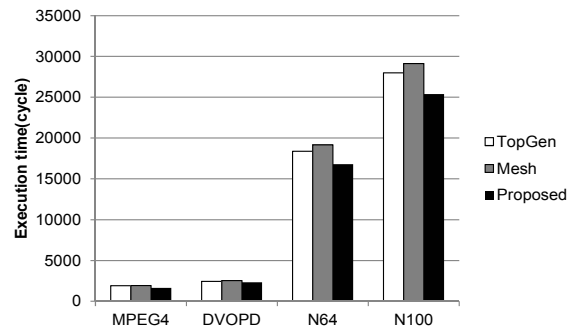
표 1은 4개의 커뮤니케이션 추적 그래프가 가지고 있 는 정보를 나타낸다.

그림 6은 DVOPD와 MPEG4, 그리고 총 64개와 100 개의 노드로 구성된 커뮤니케이션 추적 그래프에 따른 기존 정형 토폴로지의 Mesh 구조와 TopGen의 비정형 토폴로지 및 제안하는 분리된 주소/데이터 네트워크를 위한 토폴로지에서의 성능 비교 평가를 나타낸다.

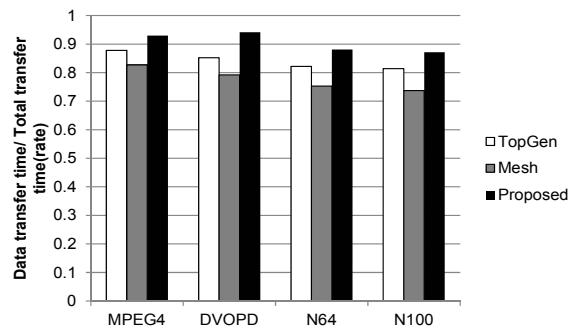
본 논문에서 제안한 방식은 DVOPD와 MPEG4에서의 평균 지연시간과 실행시간은 Mesh구조에서보다 각 각 21.99%, 11.44% 감소하였고, TopGen보다 각각 12.10%, 8.85% 감소하였다. 또한 전송 효율 면에서, 총 전송시간 대비 실질 데이터 전송 시간 비율이 Mesh구



(a)



(b)



(c)

그림 6. 커뮤니케이션 추적 그래프에 따른 성능 측정. (a) 평균 지연시간 비교 평가, (b) 실행시간 비교 평가, (c) 총 전송 시간 대비 실질 데이터 전송 시간 비율

Fig. 6. Performance evaluation under CTG. (a) Comparison on average latency, (b) Execution time, (c) Comparison on percentage of data transfer time on total transfer time.

조에서보다 15.56%, TopGen보다 8.16% 향상되었다.

또한, TGFF로 생성한 N64와 N100에서 비교 시 제 안한 방식은, Mesh구조와 비교 시 평균 지연시간과 실행시간은 각각 28.90%, 12.59% 감소하였고, TopGen에서 생성된 비정형 토폴로지에 비해 각각 15.6%, 9.94% 감소하였다. 전송 효율 면에서는 총 전송시간 대비 실질 데이터 전송 시간 비율이 Mesh보다 17.64%, TopGen에서 생성된 비정형 토폴로지 대비 7.16% 향상 되었다.

벤치마크 어플리케이션과 TGFF로 생성한 커뮤니케이션 추적 그래프의 실험결과를 비교해 보았을 때, 제안한 설계 기법은 TGFF로 생성한 복잡한 커뮤니케이션 추적 그래프에서 더욱 좋은 효율을 보인다. 즉, 제안하는 토폴로지 설계기법은 네트워크의 복잡도가 큰 어플리케이션에서 경로 충돌과 지연시간에 더 효과적임을 볼 수 있었다.

## VI. 결 론

본 논문에서는 버스 프로토콜 호환 가능한 NoC 설계에서 다단계 버스 트랜잭션 단계에서 나타나는 홉 수와 경로 충돌의 대립 관계 문제를 해결하기 위해 분리된 주소/데이터 네트워크 토폴로지 생성 기법을 제안하였다.

위의 토폴로지 생성 기법을 검증하고자 SystemC 기반의 시뮬레이터를 구축하고, 2개의 벤치마크 어플리케이션과 TGFF를 통해 생성된 복잡한 트래픽 특성을 갖는 2개의 커뮤니케이션 추적 그래프에서 실험한 결과, Mesh구조와 TopGen의 비정형 구조에서보다 평균 지연시간에서 개선된 결과를 보였다. 또한 분리된 주소/데이터 네트워크를 사용하여 데이터 전송에서의 경로 충돌 문제를 최소화하는 구조를 통해 총 전송 시간 대비 실질 데이터 전송 시간 비율이 효과적으로 증가함을 볼 수 있었다.

## REFERENCES

- [1] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures," *IEEE Trans. Computers*, Vol. 54, no. 8, pp. 1025-1040, Aug. 2005.
- [2] B. A. A. Zitouni and R. Tourki, "Design and implementation of network interface compatible OCP for packet based NoC," in *Proc. of Int. Conf. Design and Technology of Integrated Systems in Nanoscale Era (DTIS)*, pp. 1-8, Mar. 2010.
- [3] C. Ababei, "Efficient Congestion-oriented Custom Network-on-Chip Topology Synthesis," in *Proc. of Int. Conf. Reconfigurable Computing and FPGAs (ReConFig)*, pp. 352-357, Dec. 2010.
- [4] S. Deniziak and R. Tomaszewski, "Contention-avoiding custom topology generation for network-on-chip," in *Proc. of Int. Symp. Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, pp. 234-237, April 2009.
- [5] A. Shacham, K. Bergman, and L. P. Carloni, "On the Design of a Photonic Network-on-Chip," in *Proc. of Int. Symp. Networks-on-Chip (NOCS)*, pp. 53-64, May 2007.
- [6] J. Balfour and W. Dally, "Design tradeoffs for tiled CMP on-chip networks," in *Proc. of Int. Conf. Supercomputing (ICS)*, pp. 298-198, June 2006.
- [7] A. K. Mishra, O. Mutlu, and C. R. Das, "A heterogeneous multiple network-on-chip design: an application-aware approach," in *Proc. of Design Automation Conference (DAC)*, pp. 36, May 2013.
- [8] A. K. Lusala and J. D. Legat, "A hybrid router combining sdm-based circuit switching with packet switching for on-chip networks," in *Proc. of Int. Conf. Reconfigurable Computing and FPGAs (ReConFig)*, pp. 340-345, Dec. 2010.
- [9] H. Matsutani, M. Koibuchi, Y. Yamada, D. F. Hsu, and H. Amano, "Fat H-Tree: A Cost-Efficient Tree-Based On-Chip Network," *IEEE Trans. Parallel and Distributed Systems*, Vol. 20, no. 8, pp.1126-1141, Aug. 2009.
- [10] Y. Wang, Y. Pan, and X. Yan, and R. Huan, "An On-Line Reconfigurable Four-Ary Tree-Based Network on Chip for Distributed Particle Filters," in *Proc. of Int. Conf. Computer Science and Network Technology (ICCSNT)*, pp. 2102-2106, Dec. 2012.
- [11] K. Andreev, H. Racke, "Balanced Graph Partitioning," *Theory of Computing Systems*, Vol. 39, no. 6, pp. 929-939, Nov. 2006.
- [12] H. Moussa, A. Baghdadi, and M. Jequequel, "On-chip communication network for flexible multiprocessor turbo decoding," in *Proc. of Int. Conf. Information and Communication Technologies: From Theory to Applications (ICTTA)*, pp. 1-6, April 2008.
- [13] D. S. Oh and K. K. Parhi, "Low-complexity switch network for reconfigurable LDPC decoders," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, Vol. 18, no. 1, pp. 85-94, Mar. 2009.
- [14] P. K. Sahu and S. Chattopadhyay, "A survey on application mapping strategies for Network-on-Chip design," *Journal of Systems Architecture*, Vol. 59, Issue 1, pp. 66-76, Jan. 2013.
- [15] R. P. Dick, D. L. Rhodes, and W. Wolf, "TGFF: Task Graphs for Free," in *Proc. of Int. Workshop on Hardware/Software Codesign (CODES/CASHE)*, pp. 97-101, Mar. 1998.
- [16] G. Fen and W. Ning, "A Minimum-Path

Mapping Algorithm for 2D Mesh Network on Chip Architecture,” in Proc. of Asia Pacific Conf. Circuits and Systems (APCCAS), pp. 1542-1545, Nov. 2008.

[17] Y. Ar, S. Tosun, “TopGen: A New Algorithm

for Automatic Topology Generation for Network on Chip Architectures to Reduce Power Consumption,” in Proc. of Int. Conf. Application of Information and Communication Technologies (AICT), pp. 1-5, Oct. 2009.

저 자 소 개



정 승 아(학생회원)  
2014년 한국외국어대학교 전자공학과 학사 졸업.  
2014년 9월~현재 성균관대학교 전자전기컴퓨터공학과 석사 과정.  
<주관심분야: SoC 설계, NoC>



이 재 훈(학생회원)  
2011년 성균관대학교 반도체시스템공학과 학사 졸업  
2014년 성균관대학교 전자전기컴퓨터공학과 석사 졸업.  
2014년 3월~현재 성균관대학교 전자전기컴퓨터공학과 박사 과정.

<주관심분야: SoC 설계, NoC>



김 상 현(학생회원)  
2015년 성균관대학교 전자전기공학과 학사 졸업.  
2015년 3월~현재 성균관대학교 전자전기컴퓨터공학과 석사과정.  
<주관심분야: SoC 설계, NoC>



이 재 성(학생회원)  
2015년 고려대학교 전자 및 정보공학과 학사 졸업.  
2015년 3월~현재 성균관대학교 전자전기컴퓨터공학과 석박사과정.

<주관심분야: SoC 설계, NoC>



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 삼성 전자 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, D IC, 메모리/스토리지 시스템 구조, 임베디드 SW, IT 융합기술>