

Hybrid NoC 시스템을 위한 재구성 가능한 스위치 설계

준회원 이 동 열*, 정회원 황 선 영*

Design of a Dynamically Reconfigurable Switch for Hybrid Network-on-Chip Systems

Dong-Yeol Lee* Associate Member, Sun-Young Hwang* Regular Member

요 약

본 논문은 다양한 멀티미디어 어플리케이션을 수행하는 hybrid NoC 시스템을 위한 새로운 동적 재구성 가능한 스위치를 제안한다. 기존의 고정된 스위치와 job 분배 알고리즘을 사용하는 hybrid NoC 구조는 효과적인 동작을 위하여 해당 NoC 시스템에서 수행될 어플리케이션을 정확히 예측해야 한다. 본 논문은 NoC 시스템에서 수행되는 다양한 멀티미디어 어플리케이션에 대하여 버퍼 오버플로우를 최소화할 수 있는 재구성 가능한 스위치 구조를 제안한다. 제안된 시스템의 검증을 위하여 임베디드 시스템에서 사용되는 다양한 멀티미디어 어플리케이션 중 MPEG4 동영상 재생, MP3 재생, GPS 위치 계산, OFDM 복조를 대상으로 실험하였다. 버퍼 오버플로우는 단일 구조의 서브 클러스터로 mesh 토폴로지와 star 토폴로지를 갖는 NoC와 비교하여 각각 평균 41.8%와 29.0%의 감소를 보인다. 전력 소모에서는 고정된 스위치를 사용한 hybrid NoC 구조와 비교하여 평균 2.3%의 증가를 보인다. 면적에서는 서브 클러스터의 구조에 따라 -0.6% ~ 5.7%의 증가를 보인다.

Key Words : Reconfigurable, Switch, Network-on-Chip, Buffer Overflow, Hybrid Network

ABSTRACT

This paper proposes a novel dynamically reconfigurable switch for various multimedia applications in hybrid NoC systems. Current NoC systems, which adopt hybrid NoC structure with fixed switch and job distribution algorithms, require designers to precisely predict the property of applications to be processed. This paper proposes a reconfigurable switch which minimizes buffer overflow in various multimedia applications running on an NoC system. To verify the performance of the proposed system, we performed experiments on various multimedia applications running on embedded systems, such as MPEG4 and MP3 decoder, GPS positioning system, and OFDM demodulator. Experimental results show that buffer overflow has been decreased by 41.8% and 29.0%, respectively, when compared with NoC systems having sub-clusters with mesh or star topology. Power usage has been increased by 2.3% compared with hybrid NoC systems using fixed switches, and chip area has been increased from -0.6% to 5.7% depending on sub-cluster topology.

I. 서 론

Deep Sub-Micron (DSM) 공정 기술의 발전에 따라 하나의 칩에 여러 기능 모듈들이 집적되어 하

나의 시스템을 구성하는 System-on-Chip (SoC)에 대한 연구가 활발히 이루어지고 있다^[1]. 기존의 SoC에서는 칩 내의 IP들의 연결을 위해 버스 구조를 사용하였으나 DSM으로 제작된 칩에서 버스 구조는

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(IITA-2009-(C1090-0902-0012))

* 서강대학교 전자공학과 대학원 CAD & ES 연구실(hwang@sogang.ac.kr)

논문번호: KICS2009-03-125, 접수일자: 2009년 3월 23일, 최종논문접수일자: 2009년 8월 10일

물리적, 기능적 한계에 직면하였다^{2),3)}. 버스 구조는 물리적, 기능적 한계로 인하여 각 IP가 통신에서 요구하는 대역폭과 에너지 소모량, 네트워크의 크기 증가에 무관한 성능을 보장할 수 없다. 이러한 문제점을 극복하기 위해 스위치 연결을 이용한 Network-on-Chip (NoC)이 제안되었다⁴⁾. 최근에 SoC에는 많은 수의 코어와 IP블록들이 집적되고, 코어의 성능을 최대한 활용하기 위해 여러 코어와 IP블록들이 멀티 프로세싱을 수행하여, 코어 간의 데이터 통신이나 공유 메모리 접근을 위한 트래픽이 증가한다. 증가된 트래픽은 특정 네트워크 구간에 집중되는 현상을 발생시켜 네트워크 리소스의 과도한 사용을 요구한다. 트래픽이 집중된 스위치는 송수신간의 속도차이로 인해 수신된 데이터가 버퍼의 저장 용량을 초과하게 되어 스위치의 버퍼 오버플로우가 발생한다. 스위치의 버퍼 오버플로우가 발생하면, 버퍼의 데이터의 일부 혹은 전부를 폐기시킨다. 폐기된 데이터의 재송신에 따른 연산 증가와 네트워크 부하에 따른 전송 지연은 코어의 처리 속도 향상에도 불구하고 시스템 성능 증가를 기대하기 어렵게 만든다⁶⁾. NoC에서는 네트워크의 전송 지연 등의 네트워크 오버헤드를 줄이기 위한 동적 라우팅 알고리즘이 제안되었다. 동적 라우팅 방식은 데이터 전송 경로를 네트워크 상황에 따라 실시간으로 변경하여 네트워크 리소스의 점유율이 높은 경로를 대신하여 상대적으로 점유율이 낮은 경로로 데이터를 우회 전송함으로써 네트워크 리소스를 최대한 활용하면서 네트워크 전체의 throughput을 향상시킨다⁷⁾. 트래픽 예측을 통해 스위치 버퍼의 크기나 네트워크 대역폭을 미리 결정하고 이를 바탕으로 패킷 폐기 등을 최소화시킬 수 있는 효율적인 네트워크 구조에 관한 연구도 진행되었다⁸⁾.

최근 임베디드 시스템은 음성, 영상, GPS, 무선 통신 등 다양한 멀티미디어 응용의 멀티프로세싱을 요구한다⁹⁾. 임베디드 시스템에 사용되는 NoC 시스템은 다양한 멀티미디어 응용의 동작 특성에 따른 구조와 네트워크 알고리즘을 고려할 필요가 있다. PDA와 같이, 임베디드 시스템을 이용한 디바이스는 다양한 사용자 어플리케이션을 구동시켜주는 추세이다. 임베디드 시스템을 설계하는 과정에서 해당 칩이 어떠한 어플리케이션을 구동시켜줄지를 예측하여 사전에 토폴로지를 결정하는 것은 어렵기 때문에, hybrid 네트워크 토폴로지를 이용한 job 분배 알고리즘이 제안되었다¹⁰⁾. 클러스터링을 통해 로컬 네트워크와 글로벌 네트워크를 다양한 토폴로지

구축하고, 어플리케이션 특성과 네트워크 토폴로지 사이의 상관관계를 이용해 스위치 버퍼의 오버플로우를 최소화하는 알고리즘을 제안한다. 각각의 어플리케이션에 대하여 단일 토폴로지에서의 실험을 수행한 결과에 따라 서브 클러스터의 최적 토폴로지 구조를 찾고, 클러스터링을 통해 글로벌 네트워크와의 연결을 구성한다.

본 논문은 다음과 같이 구성된다. 2절에서는 기존의 제안된 NoC의 특징을 설명하고, 3절에서는 논문에서 제안된 재구성 가능한 스위치의 특징과 동적 job 분배 알고리즘에 대하여 설명한다. 4절에서는 패킷 폐기율, 전력 소모와 면적에 대한 실험 결과를 보인다. 마지막으로 5절에서는 결론을 제시한다.

II. 관련 연구

MP-SoC 시스템에서 NoC가 온-칩 연결구조 아키텍처의 해결책으로 제안된 이후, NoC 아키텍처는 다양한 통신 제약 조건을 만족시키기 위해 활발히 연구가 진행되었다^{11),12)}. 기존의 연구에서는 버퍼의 오버플로우를 해결하기 위한 동적 라우팅 기법, 버퍼 최적화 기법, hybrid 네트워크를 이용한 Job 분배 알고리즘이 제안되었다^{7),8),10)}. 2.1절에서는 여러 코어들과 IP들 간의 네트워크 성능 개선을 위해 기존의 연구에서 제안된 NoC에 대해 설명하고, 2.2절에서는 기존에 제안된 대역폭 확대 가능한 크로스바 스위치에 대해 설명한다. 2.3절에서는 어플리케이션과 NoC 네트워크 토폴로지 간의 상관관계를 통해 연구한 job 분배 알고리즘에 대해 설명한다.

2.1 Network-on-Chip

SoC 시스템에서 증가한 코어와 IP 들간의 트래픽에 따라 기존의 버스 구조로 트래픽을 고속으로 처리하기에 한계에 이르렀다. 증가된 트래픽을 성능저하 없이 전송하기 위해 다중 레이어 방식의 버스를 사용하기도 하나, 버스 사용 권한을 획득하기 위한 arbitration에 의한 오버헤드 등으로 인해 증가된 트래픽을 시스템의 성능저하 없이 처리하기 어렵다. 이에 대한 해결책으로 제안된 NoC는 IP 코어간의 안정적인 통신, 오류 정정, 높은 확장성 등의 장점으로 인해 기존의 SoC 버스의 대안으로 제안되었다¹³⁾.

다량의 데이터를 빠른 속도로 전달하기 위해서는 wire의 대역폭을 키워야 한다. 대역폭의 증가는 wiring에 따른 면적과 전력 소비가 증가하여 NoC

시스템이 주로 사용되는 임베디드 시스템에서는 단점으로 작용한다. NoC 시스템에서 네트워크 구조는 패킷 전달 속도와 시스템 면적에 직접적인 영향을 주므로 패킷 전달에 효율적이면서 면적을 최소화할 수 있는 네트워크 구성이 요구되며, 트래픽이 일부 네트워크 리소스에 집중되어 패킷 전달 불능에 빠지지 않도록 트래픽을 분산시키는 알고리즘과 구조에 대한 연구가 요구된다.

2.2 대역폭 확대 가능한 크로스바 스위치

Adaptive한 대역폭을 지원하기 위해 추가적인 버스를 이용한 재구성 가능한 크로스바 스위치가 제안되었다^[4]. 크로스바 스위치의 각 입력 포트는 다른 빈도로 동작하는 IP에 의해 다른 대역폭을 요구한다. 기존의 크로스바 스위치는 각 포트의 대역폭이 일정하기 때문에 순간적으로 트래픽이 집중되는 입력 포트에서 성능저하를 초래하였다. 각 포트의 최대 요청 대역폭에 맞추어 크로스바 스위치를 설계하면, 많은 시간대에서 대역폭의 낭비를 가져온다. 이러한 문제를 해결하기 위해 기존의 크로스바 스위치에 추가적으로 공유 버스를 구성하는 스위치 구조를 제안하였다^[4]. 입력 포트 중에 트래픽이 집중되는 포트에 기존의 크로스바 스위치와 공유 버스를 모두 사용하게 하여 빠른 트래픽 처리를 하며, 다수의 입력 포트에서 공유 버스의 사용권한을 요청할 경우에는 라운드-로빈(round-robin)방식에 따라 버스의 사용권한을 부여한다. 이 연구는 크로스바 스위치 내부 경로의 대역폭만 고려하였다.

2.3 Hybrid topology NoC를 위한 Job 분배 알고리즘

어플리케이션의 특성에 따라 각각 mesh 혹은 star 토폴로지로 이루어진 서브 클러스터와 글로벌 네트워크의 hybrid 토폴로지 구조와 이를 지원하기 위한 job 분배 알고리즘이 제안되었다^[10]. Hybrid topology는 4개의 코어로 구성된 서브 클러스터가 4개 존재하여 총 16개의 코어로 구성되며, job 분배 알고리즘의 적용을 위해 LUT (Look Up Table)에는 각 어플리케이션에 적합한 토폴로지 정보가 저장된다. Job 분배 알고리즘은 LUT의 정보를 바탕으로 패킷 폐기를 최소화할 수 있는 후보 클러스터들을 선정하여 후보 클러스터들 중에 idle 상태의 클러스터가 있을 경우 job을 할당한다. 후보 클러스터들 중에 idle 상태의 클러스터가 없을 때 후보 클러스터들 중 버퍼의 사용률이 적은 클러스터에 job을 할당한다.

다. 적절한 클러스터를 찾지 못한 경우 라운드-로빈 방식으로 job을 할당한다. 사용자에게 의한 추가적인 프로그램 설치가 가능한 장치들이 증가하는 환경에서 이 알고리즘은 서브 클러스터와 글로벌 네트워크가 고정된 토폴로지를 갖는다. 칩 생산 단계에서 효과적인 토폴로지를 예측하기 어렵기 때문에 다양한 서브 클러스터 조합의 칩을 갖추어야 하는 단점이 있다.

III. 제안된 재구성 가능한 스위치

본 장에서는 제안된 재구성 가능한 스위치 구조에 대해 기술한다. 3.1 절에서는 재구성 가능한 hybrid NoC 토폴로지를, 3.2 절에서는 각 스위치의 구조를 설명한다. 3.3 절에서는 보낸 job 분배 알고리즘을 제시한다.

3.1 재구성 가능한 Hybrid NoC 토폴로지

기존의 hybrid NoC 아키텍처는 각 서브 클러스터와 글로벌 네트워크에 고정된 토폴로지를 사용하였다. 고정된 토폴로지로 설계하기 위해 최적화된 토폴로지를 예측하는 어려움이 있다. 이러한 어려움을 해결하기 위해 그림 1에 제시한 재구성 가능한 hybrid NoC 토폴로지를 제안한다. 스위치는 구조에 따라 Edge switch (E_SW), Corner switch (C_SW), Internal switch (I_SW), Star switch (S_SW)로 분류한다.

그림 1에서 각 서브 클러스터는 4개의 코어로 구성되고 글로벌 네트워크에 4개의 서브 클러스터가 존재한다. 각 서브 클러스터와 글로벌 네트워크는 mesh 토폴로지와 star 토폴로지를 위한 연결 구조를 가지고

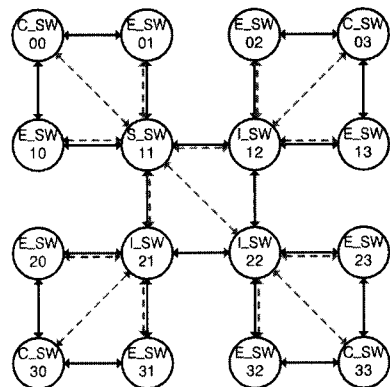


그림 1. 재구성 가능한 hybrid Network-on-Chip 토폴로지

있으며, 16 비트 full-duplex 연결을 지원한다. Mesh 토폴로지와 star 토폴로지를 모두 동작시키기 위한 추가적인 스위치를 없애기 위해 전체 네트워크는 4 종류 (Edge switch, Corner switch, Internal switch, Star switch)의 스위치로 구성되며, 글로벌 네트워크의 통신은 Internal switch와 Star switch에 의해 이루어진다. 각 스위치의 구조는 3.2절에서 설명한다.

영상 신호 처리(MPEG4)와 오디오 신호 처리(MP3)는 mesh 토폴로지에서도 버퍼 오버플로우가 적게 발생하며, GPS 위치 연산과 OFDM 복조에서는 star 토폴로지에서도 버퍼 오버플로우가 적게 발생한다^[10]. Job 분배기에서 해당 어플리케이션에 따라 서버 클러스터와 글로벌 네트워크의 토폴로지를 결정하고 job을 최적화된 토폴로지의 코어에 할당한다.

본 연구에서 구현한 NoC 시스템의 각 코어는 2 비트의 서버 클러스터 내 주소를 갖고, 다른 서버 클러스터로 전송될 경우 각 서버 클러스터의 gateway에서 2 비트의 서버 클러스터 주소를 덧붙인다. 전송하고자 하는 패킷의 목적지가 서버 클러스터 내 부인지 외부인지를 구별하기 위하여 1 비트의 OFlag를 사용하며, OFlag 비트가 1인 경우 뒤에 따르는 4 비트를 목적지 주소로 하여 다른 서버 클러스터로 패킷을 전송한다. 패킷의 크기는 60 바이트이고 각 버퍼의 크기는 180 바이트이다.

3.2 스위치의 구조

Mesh 토폴로지와 star 토폴로지 간에 재구성 가능한 네트워크를 위해 각 스위치는 컨트롤 신호로 2비트의 Topology_sel을 사용한다. Topology_sel[0]은 서버 클러스터의 토폴로지를 결정하고 Topology_sel[1]은 글로벌 네트워크의 토폴로지를 결정한다. 두 비트의 Topology_sel 신호는 공통으로 0 이면 mesh 토폴로지, 1 이면 star 토폴로지의 연결을 갖는다. NoC에서 면적과 전력 소모에서 가장 큰 비중을 차지하는 요소는 버퍼이다^{[15],[16]}. 그림 1의 전체 네트워크 구조에서 개별 스위치의 경우 mesh 토폴로지 동작보다 star 토폴로지 동작에서 필요로 하는 버퍼블록의 수가 감소한다. Star 토폴로지 동작에서 mesh 토폴로지에서도 필요로 하였던 입출력 버퍼 중 일부만 사용하면 star 토폴로지 동작 시 면적과 전력 소모에서 낭비를 가져온다. 동일한 통신환경에서 버퍼의 크기가 증가할 경우 버퍼 오버플로우의 발생 빈도는 감소한다. 재구성 가능한 스위치의 star 토폴로지 동작 시 낭비되는 버퍼를 줄이기 위해 효율적인 스위치 구조를 제안한다. 그림 2는 Corner

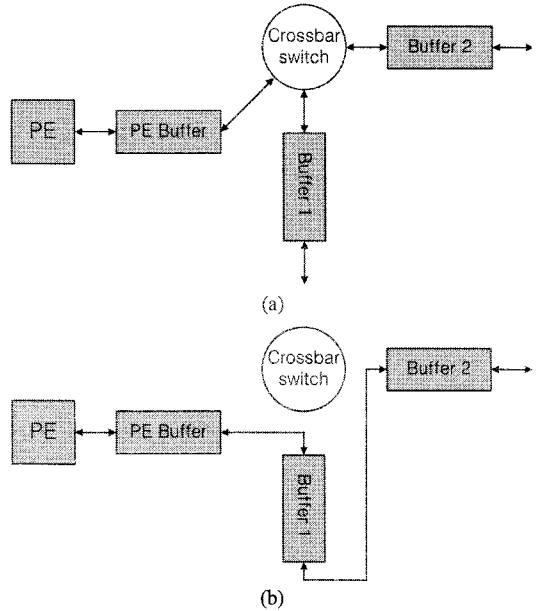


그림 2. Corner switch와 Edge switch의 버퍼 연결
(a) L-Mesh, (b) L-Star

switch와 Edge switch의 버퍼 활용을 최대한으로 하기 위한 토폴로지에 따른 버퍼 연결을 보인다.

Corner switch와 Edge switch는 mesh 토폴로지와 star 토폴로지 모두 동작 시, PE (Processing Element) 방향 외에 각각 두 방향과 한 방향의 포트를 필요로 한다. Corner switch는 star 토폴로지 동작 시 mesh 토폴로지와는 다른 방향의 개별적인 wire를 요구한다. Corner switch의 mesh 토폴로지와 star 토폴로지의 개별 wire 선택은 Mux와 Demux에 의해 이루어진다. Edge switch는 star 토폴로지 동작 시 mesh 토폴로지와 동일한 포트를 사용하기 때문에 추가적인 wire를 요구하지 않는다. Corner switch와 Edge switch의 동작은 서버 클러스터의 토폴로지에만 영향을 받기 때문에 컨트롤 신호로 Topology_sel[0] 신호만 사용된다. Mesh 토폴로지와 star 토폴로지에 대하여 최대한으로 필요한 버퍼의 수는 mesh 토폴로지의 경우가 많기 때문에 전체 버퍼의 수는 mesh 토폴로지에서도 필요로 하는 만큼 갖도록 한다. 그림 2(b)와 같이 star 토폴로지 동작 시에는 내부의 크로스바 스위치를 거치지 않고 Mux와 Demux에 의하여 PE에서 PE Buffer, Buffer 1, Buffer 2를 거쳐 스위치 외부의 네트워크와 연결되며 그림 1의 Star switch 또는 Internal switch에 의하여 라우팅이 이루어진다. Corner switch와 Edge switch의 버퍼 동작은 서버 클러스터가

star 토폴로지로 동작 시에 스위치 외부와의 통신에서 Corner switch에서는 mesh 토폴로지와 다른 방향의 연결을 갖는다는 점을 제외하면 동일하다.

Internal switch와 Star switch는 mesh 토폴로지와 star 토폴로지로 동작 시 각각 4 방향의 포트를 필요로 한다. Star 토폴로지로 동작 시에는 mesh 토폴로지와는 다른 방향의 개별적인 wire를 요구한다. 스위치의 동작은 서버 클러스터와 글로벌 네트워크 토폴로지 모두에 영향을 받기 때문에 컨트롤 신호로 Topology_sel[0]와 Topology_sel[1]의 두 비트가 사용된다. Internal switch의 버퍼 활용을 최대한으로 하기 위한 토폴로지에 따른 버퍼의 연결을 그림 3에 제시한다.

그림 3(b)는 Internal switch의 글로벌 네트워크와 서버 클러스터가 각각 mesh 토폴로지와 star 토폴로지로 동작할 때의 버퍼 사용을 보인다. 서버 클러스터가 star 토폴로지로 동작할 때 Edge switch나

Corner switch에 있는 PE와의 통신에는 Edge switch와 Corner switch의 버퍼만을 사용한다. 글로벌 네트워크의 mesh 방향 버퍼 외에 여유가 있는 두 개의 버퍼를 PE에서 사용한다. PE가 PE 버퍼 외에 추가적인 두 개의 버퍼를 사용하여 서버 클러스터의 PE와 동일한 조건의 버퍼 사용을 가질 수 있으며, 버퍼 크기 증가로 버퍼 오버플로우를 감소시킬 수 있다. 그림 3(c)는 Internal switch의 글로벌 네트워크와 서버 클러스터가 각각 star 토폴로지와 mesh 토폴로지로 동작할 때의 버퍼 사용을 보인다. 서버 클러스터의 mesh 토폴로지에 사용되는 두 방향의 포트 외에 남는 두 버퍼를 모두 글로벌 네트워크의 star 토폴로지에 사용한다. 그림 3(d)는 Internal switch의 글로벌 네트워크와 서버 클러스터가 모두 star 토폴로지로 동작할 때의 버퍼 사용을 보인다. Edge switch와 Corner switch는 서버 클러스터가 star 토폴로지로 동작 시에 PE와 네트워크와

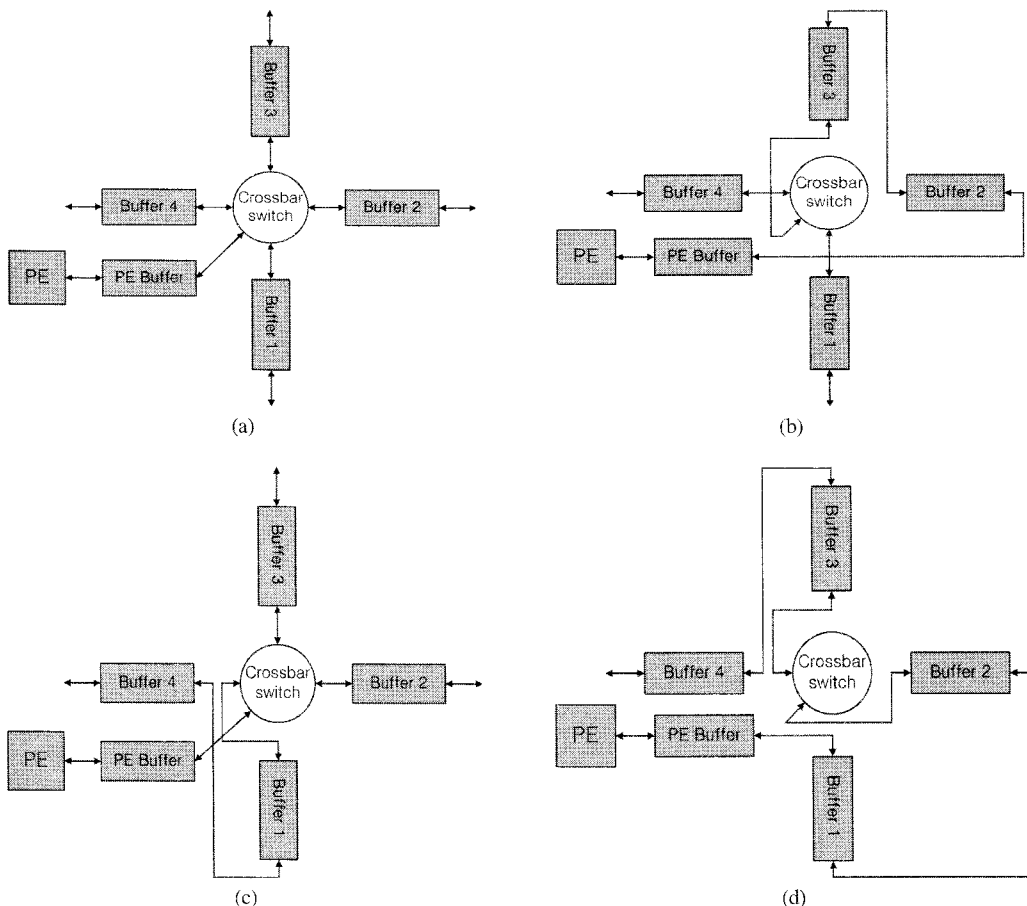


그림 3. Internal switch의 버퍼 연결 (a) G-Mesh/L-Mesh, (b) G-Mesh/L-Star, (c) G-Star/L-Mesh, (d) G-Star/L-Star

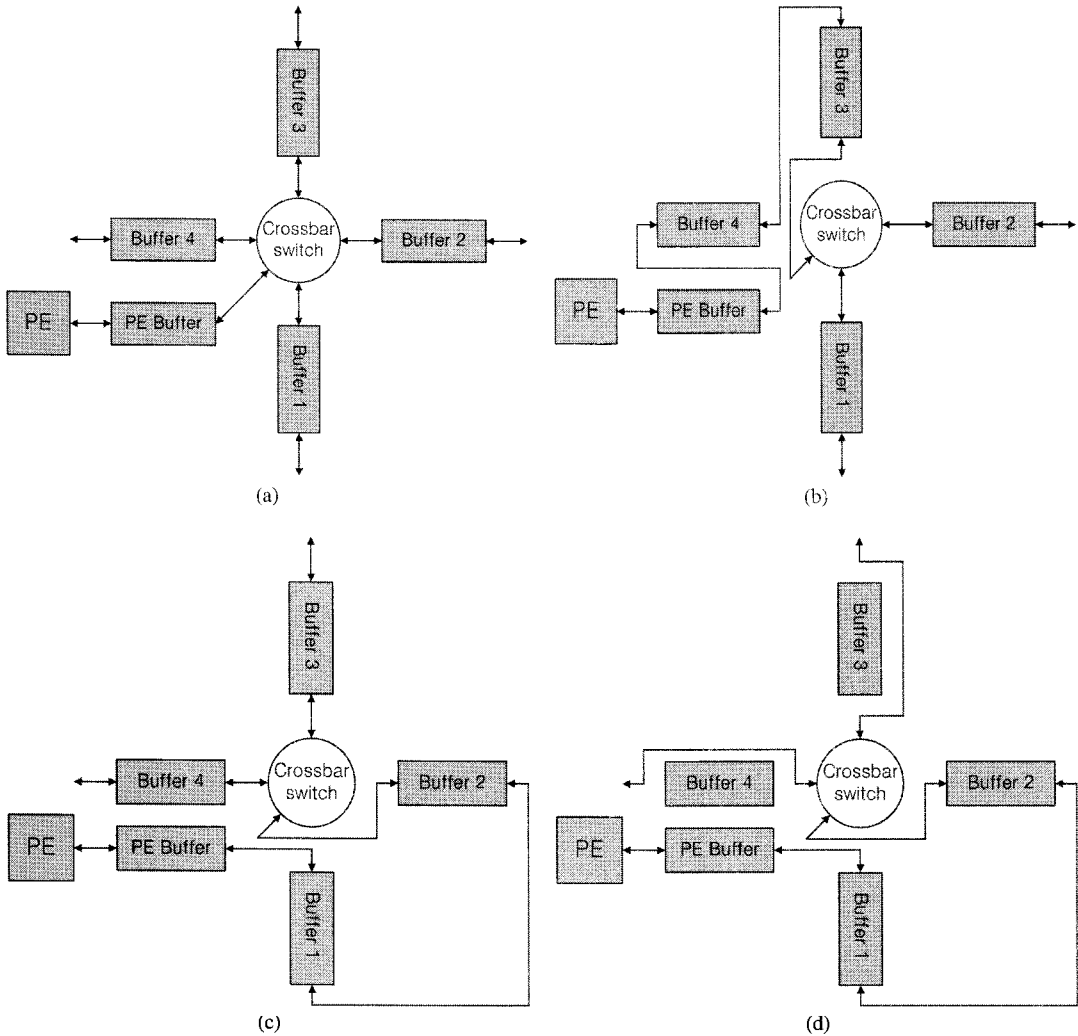


그림 4. Star switch의 버퍼 연결 (a) G-Mesh/L-Mesh, (b) G-Mesh/L-Star, (c) G-Star/L-Mesh, (d) G-Star/L-Star

의 연결에서 PE 버퍼 외에 각각 2개의 버퍼를 추가로 사용한다. Internal switch의 PE에도 동일한 조건을 위해 두 개의 추가적인 버퍼를 사용한다. 남은 두 개의 버퍼는 글로벌 네트워크와의 통신에 사용한다. Star switch의 버퍼 활용을 최대한으로 하기 위한 토폴로지에 따른 버퍼의 연결을 그림 4에 제시한다.

그림 4(d)의 글로벌 네트워크와 서브 클러스터가 모두 star 토폴로지를 갖는 경우를 제외한 버퍼 연결은 Internal switch의 경우와 비슷하다. 그림 4(d)는 Star switch의 글로벌 네트워크와 서브 클러스터가 모두 star 토폴로지로 동작할 때의 버퍼 사용을 보인다. Star 토폴로지 동작에서 서브 클러스터의 스위치와 동일한 조건을 위해 PE와 네트워크와의 연결에서 PE 버퍼 외에 추가적으로 2개의 버퍼를

사용하고, 남은 두 개의 버퍼는 전체 네트워크의 데이터 전달 속도의 균형을 맞추기 위해 사용하지 않는다.

3.3 동적 Job 분배 알고리즘

임베디드 시스템에 주어지는 job에 대하여 버퍼 오버플로우가 최소가 되는 최적의 서브 클러스터로 job을 맵핑하기 위한 동적 job 분배 알고리즘을 제안한다. Job 분배기는 수행될 각각의 멀티미디어 어플리케이션에 대해 최적으로 수행되는 토폴로지에 대한 정보가 있는 LUT를 가지고 있으며 job은 자신이 포함된 어플리케이션에 대한 정보 비트를 가지고 있다^[10]. Job 분배기는 각 서브 클러스터에 대하여 버퍼 사용률을 나타내기 위하여 2 비트의 flag

값을 갖는다. Flag 값 '00'은 버퍼 사용률이 0 %, '10'은 70 % 이하를 나타낸다. '11'은 버퍼 사용률이 70 %를 초과함을 나타낸다. 제안하는 동적 job 분배 알고리즘은 어플리케이션 정보에 따라 적절한 토폴로지를 설정하고 해당 서버 클러스터에 job을 할당한다. Job 분배기의 동작은 다음과 같다. LUT를 통하여 버퍼 오버플로우를 최소화하는 토폴로지를 정하고 idle 상태의 서버 클러스터가 있을 경우 Topology_sel 신호를 통해 토폴로지를 결정한 후 job을 분배한다. Idle 상태의 서버 클러스터가 없을 경우에는 참고문헌 [10]에서 제시한 job 분배 알고리즘 step 4 이후의 과정을 따른다.

IV. 실험 결과

제안된 NoC 아키텍처는 verilogHDL을 이용하여 RTL 레벨에서 모델링하였다. 면적과 전력 소모량 측정을 위해 Synopsys 사의 design compiler와 TSMC 사의 TCBN90GHPBC(90nm) 공정 라이브러리를 사용하여 합성하였으며^[17], 정확한 동적 파워 예측을 위해서 Synopsys 사의 power compiler를 사용하였다^[18]. 실험은 임베디드 시스템에서 많이 사용되는 MPEG4, MP3, GPS 위치 계산, OFDM 복조를 이용하여 수행하였다. MPEG4와 MP3는 mesh 토폴로지에서 패킷 폐기율이 낮게 나타나고, GPS 위치 계산과 OFDM 복조는 star 토폴로지에서 패킷 폐기율이 낮게 나타난다^[10]. 실험에 사용된 영상 데이터는 720*480 해상도의 MPEG4 코덱으로 인코딩되었으며, 음성 데이터는 128 Kbps 수준으로 압축된 MPEG1 audio layer-3 코덱으로 인코딩되었다. GPS 위치 데이터는 4채널 방식의 송신기를 기준으로 프레임당 1.5 Kb로 50 bit/sec 전송율을 기준으로 설계하였다. OFDM 복조는 MPEG4에서 사용한

데이터를 미리 OFDM 방식으로 변조 후 사용하였다. 정확한 실험 결과를 위해 종류별로 각 10개씩의 데이터에 대해 반복적으로 실험한 결과의 평균을 실험 결과로 사용하였다. 본 실험에서는 버퍼 오버플로우에 의한 패킷 폐기를 측정하기 위하여 ACK에 의한 데이터의 재전송은 없다. 표 1은 기존의 제안된 NoC 복합 구조와 본 연구에서 제안하는 NoC 구조의 패킷 폐기율 비교를 보인다.

실험에 사용된 NoC 구조는 4개의 서버 클러스터를 갖고 글로벌 네트워크가 star 토폴로지로 연결되어 있다. 실험 비교 대상의 서버 클러스터는 모두 mesh 토폴로지 또는 star 토폴로지를 가진 NoC 구조와 수행 프로그램에 따른 hybrid 서버 클러스터 구조를 가지고 있고 job 분배 알고리즘을 적용한 NoC 구조를 사용하였다^[10]. 실험에 사용한 어플리케이션 조합은 "MPEG4 + MP3 + GPS + OFDM", "2개의 MPEG4 + MP3 + GPS", "MP3 + 2개의 OFDM + GPS"이다. 이는 각각 "2개의 mesh 토폴로지 + 2개의 star 토폴로지", "3개의 mesh 토폴로지 + 1개의 star 토폴로지", "1개의 mesh 토폴로지 + 3개의 star 토폴로지"에서 최적으로 수행되는 프로그램 조합이다. 고정된 스위치를 갖는 G-Star/L-Hybrid 구조의 서버 클러스터는 각각의 어플리케이션에 맞추어 "2-mesh + 2-star", "3-mesh + 1-star", "1-mesh + 3-star" 토폴로지를 사용하였다. 제안된 구조는 단일 구조의 서버 클러스터를 가진 NoC와 비교하여 평균 -41.8%와 -29.0%의 패킷 폐기율 감소를 보인다. G-Star/L-Hybrid 구조와의 비교에서도 패킷 폐기율 증가가 평균 1.5%로 작게 나타났다. 고정된 스위치를 갖는 G-Star/L-Hybrid 구조의 NoC와 비교하여 제안된 구조는 글로벌 네트워크를 전담하는 스위치가 없다. 제안된 구조는 star 토폴로지로 동작 시 기존에 비하여 스위치 버퍼의 크기가

표 1. 기존 NoC 구조와 제안된 NoC 구조의 패킷 폐기율 비교

수행 프로그램 \ 구조	G-Star/L-Mesh	G-Star/L-Star	G-Star/L-Hybrid without reconfigurable switch ^[10]	제안된 구조 (with reconfigurable switch)
MPEG4 + MP3 + GPS + OFDM	12.3	9.8	6.7	6.9 (-42.3%/-27.6%/+3.0%)*
2*MPEG4 + MP3 + GPS	10.6	11.5	7.5	7.7 (-26.4%/-32.2%/+2.7%)*
MP3 + 2*OFDM + GPS	13.8	8.7	6.1	6.2 (-55.8%/-29.9%/+1.6%)*
평균	12.2	10.0	6.8	6.9 (-41.8%/-29.0%/+1.5%)*

*비교 대상과 제안된 구조의 패킷 폐기율 증감을 나타낸다.

표 2. 기존 NoC 구조와 제안된 NoC 구조의 전력 소모(mW) 비교

수행 프로그램 \ 구조	G-Star/L-Mesh	G-Star/L-Star	G-Star/L-Hybrid without reconfigurable switch ^[10]	제안된 구조 (with reconfigurable switch)
MPEG4 + MP3 + GPS + OFDM	564.4	479.7	316.6	324.2 (-42.6%/-32.4%/+2.3%)*
2*MPEG4 + MP3 + GPS	528.6	513.7	302.9	308.1 (-41.7%/-40.0%/+1.7%)*
MP3 + 2*OFDM + GPS	585.1	462.3	328.3	337.7 (-42.3%/-27.0%/+2.8%)*
평균	559.4	485.2	315.9	323.3 (-42.2%/-33.4%/+2.3%)*

*비교 대상과 제안된 구조의 전력 소모 증감을 나타낸다.

증가한다.

표 2는 기존의 제안된 NoC 복합 구조와 본 연구에서 제안하는 NoC 구조의 전력 소모 비교를 제시한다. 제안된 구조는 단일 구조의 서브 클러스터를 가진 NoC와 비교하여 평균 42.2%와 33.4%의 전력 소모 감소를 보인다. 수행 프로그램의 정확한 예측을 가정하고 고정된 구조의 스위치를 사용한 서브 클러스터를 갖는 G-Star/L-Hybrid 구조와 비교에서는 평균 2.3%의 전력 소모 증가를 보인다. 서브 클러스터가 고정된 형태의 Hybrid 구조로 정의된 경우와 비교하여 전력 소모에서 큰 overhead가 따르지 않으며, 각 어플리케이션 조합에 대하여도 전력 소모 증가에서 큰 차이가 없다.

표 3은 기존에 연구된 고정된 형태의 스위치를 사용하여 서브 클러스터를 구성한 Hybrid NoC 구조와 제안된 NoC 구조의 면적을 비교를 제시한다.

고정된 형태의 스위치를 사용하는 경우에는 서브 클러스터의 구조가 변화하기 때문에 각 경우에 대하여 면적이 변화한다. 제안된 구조에서는 토폴로지가 컨트럴 신호 두 비트에 따라 연결 구조만 변화하기 때문에 서브 클러스터 구조의 변화에 따른 면적 변화가 없다. 면적은 서브 클러스터가 “2-star + 2-mesh”와 “3-star + 1-mesh”의 구조를 취할 때에는 고정된 스위치를 가진 G-Star/L-Hybrid 구조에 비하여 각각 3.8%, 5.7% 증가하고, “1-star + 3-mesh”

의 경우에 비하여 0.6% 감소한다. 제안된 구조는 고정된 스위치를 사용하는 구조에 비하여 Mux와 Demux 회로에 의한 면적 증가 요소가 있지만, 글로벌 네트워크를 위한 추가적인 스위치가 존재하지 않아 전체적인 면적 증가는 무시할 정도이다^[10].

V. 결론

본 연구는 재구성 가능한 스위치와 job 분배 알고리즘을 적용한 NoC 구조를 제안한다. 기존의 고정된 스위치를 사용한 hybrid NoC 구조에서는 다양한 멀티미디어 어플리케이션에 효과적으로 대응하기 위하여 해당 칩에서 수행될 어플리케이션에 대한 정확한 예측이 이루어지거나 다양한 구조의 서브 클러스터로 설계된 칩들을 갖추어야 한다. 고정된 스위치와 멀티미디어 어플리케이션과 토폴로지 사이의 연관성에 따른 job 분배 알고리즘을 적용한 연구 결과^[10]와 비교하여, 제안된 NoC 구조는 재구성 가능한 스위치를 이용하여 하나의 NoC 구조로 다양한 멀티미디어 어플리케이션에 대하여 효과적으로 대응할 수 있다. 기존의 고정된 스위치를 갖는 NoC 구조와의 버퍼 오버플로우, 전력 소모, 면적의 비교를 위하여 임베디드 시스템에서 많이 사용되는 멀티미디어 어플리케이션인 MPEG4, MP3, GPS 위치 계산, OFDM 복조를 이용하여 검증하였다. 본

표 3. 기존 NoC 구조와 제안된 NoC 구조의 면적(mm²) 비교

서브 클러스터 구조	G-Star/L-Hybrid without reconfigurable switch ^[10]	제안된 구조 (with reconfigurable switch)
2-Star/2-Mesh	34.2	35.5 (+3.8%/+5.7%/-0.6%)*
3-Star/1-Mesh	33.6	
1-Star/3-Mesh	35.7	

* 비교 대상과 제안된 구조의 면적 증감을 나타낸다.

연구에서 제안하는 NoC 구조는 고정된 스위치 형태를 갖는 NoC 구조와 비교하여 버퍼 오버플로우와 전력 소모의 비교에서 각각 평균 1.5%, 2.3%의 작은 증가를 보였다. 면적에서는 서버 클러스터의 구조에 따라 3.8%, 5.7%의 증가와 0.6%의 감소를 보였다. 이는 전체 시스템에 비추어 볼 때 미미한 수준의 증감이다. 기존의 대부분 NoC 연구에서는 수행 어플리케이션의 트래픽 특성에 대한 고려가 없다. 수행 어플리케이션의 트래픽 특성이 토폴로지와의 연관이 있음을 제시하고 job 분배 알고리즘을 적용한 연구 결과가 제시되었으나^[10], 고정된 스위치 구조를 사용하여 다양한 어플리케이션에 대한 대처가 불충분하다. 본 연구에서는 재구성 가능한 스위치 구조와 이를 위한 job 분배 알고리즘을 제안하여 토폴로지에 의한 성능상의 제약을 감소시킨다. 제안하는 스위치는 NoC 구조에서 면적, 전력소모에서 큰 비중을 차지하는 버퍼의 효율적인 사용으로 패킷 폐기율을 최소화한다.

본 연구는 각각의 어플리케이션과 토폴로지 사이의 관계를 토대로 진행되었다. 추후에 서버 클러스터 사이에 일어나는 통신의 유형과 관련한 트래픽 패턴에 관한 연구를 통하여 글로벌 네트워크에 대한 효과적인 토폴로지를 연구할 필요가 있다.

참 고 문 헌

[1] W. Dally and B. Towles, "Route Packets, not Wires: On-Chip Interconnection Networks," in Proc. Design Automation Conf., Las Vegas, NV. pp.684-689, Nov., 2001.

[2] R. Ho, K. Mai, and M. Horowitz, "The Future of Wires," Proceeding of the IEEE, Vol.89, No.4, pp.490-504, Apr., 2001.

[3] R. Ho, K. Mai, and M. Horowitz, "The Future of Wires," Proceeding of the IEEE, Vol.89, No.4, pp.490-504, Apr., 2001.

[4] S. Kumar et al, "A Network on Chip architectures and Design Methodology," in Proc. ISVLSI, Pittsburgh, PA, pp.117-124, Apr., 2002.

[5] D. Sylvester and K. Keotzer, "Getting to the Bottom of Deep Submicron," in Proc. ICCAD, San Jose, CA, pp.203-211, Nov., 1998.

[6] W. Vanderbauwhede and D. Harle, "Architecture, Design, and Modeling of the

OPSnet Asynchronous Optical Packet Switching Node," Journal of Lightwave Technology, Vol.23, No.7, pp.2215-2228, July., 2005.

[7] M. Ali, M. Welzl, and S. Hellebrand, "A Dynamic Routing mechanism for Network on Chip," in Proc. IEEE NORCHIP, Oulu, Finland, pp.70-73, Nov., 2005.

[8] S. Manolache, P. Eles, and Z. Peng, "Buffer Space Optimization with Communication Synthesis and Traffic Shaping for NoCs," in Proc. DATE, Munich, Germany, pp.718-723, Mar., 2006.

[9] F. Karim, A. Mellan, A. Nguyen, U. Aydonat, and T. Abdelrahman, "A Multilevel Computing Architecture for Embedded Multimedia Applications," IEEE Micro, Vol.24, No.3, pp.56-66, May/June, 2004.

[10] W. Kim and S. Hwang, "Design of an Area-Efficient and Low-Power NoC Architecture Using a Hybrid Network Topology," IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E91-A, No.11, pp.3297-3303, Nov., 2008.

[11] S. Kumar et al, "A Network on Chip architectures and Design Methodology," in Proc. ISVLSI, Pittsburgh, PA, pp.117-124, Apr., 2002.

[12] L. Benini and G. De Micheli, "Network on Chip: A New SoC Paradigm," IEEE Computer, Vol.35, No.1, pp.70-78, Jan., 2002.

[13] L. Benini and G. De Micheli, *NetworksonChips: TechnologyandTools*, MorganKaufmann,2006.

[14] D. Kim, K. Lee, S. Lee, and H. Yoo, "A Reconfigurable Crossbar Switch with Adaptive Bandwidth Control for Network-on-Chip," in Proc. Int. Symp. Circuit and Systems, pp.2369-2372, May, 2005.

[15] X. Chen and et al, "Leakage Power Modeling and Optimization in Interconnection Networks," in Proc. Int. Symp. Low Power Electronics and Design, Seoul, pp.90-95, Aug., 2003.

[16] M. Coenen, S. Murali, A. Ruadulescu, K. Goossens, and G. De Micheli, "A Buffer-sizing Algorithm for Networks on Chips Using TDMA and Credit-Based End-to-end Flow Control," in

Proc. International Conference on Hardware Software Codesign, Seoul, pp.130-135, Oct., 2006.

[17] Synopsys, *DesignAnalyzer™UserGuide*, Synopsys®, June 2002.

[18] Synopsys, *PowerCompiler™UserGuide*, Synopsys®, June 2007.

이 동 열 (Dong-Yeol Lee)

준회원



2008년 2월 서강대학교 전자공학
학과

2008년 3월~현재 서강대학교
전자공학과 대학원 CAD &
ES 연구실 석사과정

<관심분야> Network on a chip
architecture design, System

on a chip design methodology

황 선 영 (Sun-Young Hwang)

정회원



1976년 2월 서울대학교 전자공
학과

1978년 2월 한국 과학원 전기
및 전자공학과 공학석사 취득

1986년 10월 미국 Stanford 대
학 전자공학 박사학위 취득

1976년~1981년 삼성 반도체 주
식회사 연구원 팀장

1986년~1989년 Stanford 대학 Center for Integrated
System 연구소 책임 연구원

1989년~1992년 삼성전자(주) 반도체 기술 자문

2002년 4월~2004년 2월 서강대학교 정보통신대학
원장

1989년 3월~현재 서강대학교 전자공학과 교수

<관심분야> CAD 및 임베디드 시스템 설계, NoC/
SoC 시스템 설계, HW/SW co-design, DSP/VLSI
시스템 설계