

논문 2011-48SD-8-5

하이브리드 광학 네트워크-온-칩에서 병렬 라우팅에 관한 연구

(A Study on the Parallel Routing in Hybrid Optical Networks-on-Chip)

서 정 택*, 황 용 중*, 한 태 희**

(Jung Tack Seo, Yong Joong Hwang, and Tae Hee Han)

요 약

네트워크-온-칩(Networks-on-Chip, NoC)은 고도로 복잡해지고 있는 다중 프로세서 시스템-온-칩(Multi-Processor System-on-chip, MPSoC)에서의 버스 트래픽 문제를 해결할 핵심기술이나 전통적인 전기적 상호 연결 구조로는 머지않아 대역폭 및 전력소비 등의 한계에 직면할 것으로 예상된다. 이러한 문제를 해결하기 위해 광학적 상호연결과 전기적 상호연결을 같이 사용하는 하이브리드 광학 NoC기술이 최근 활발히 연구되고 있다. 대부분의 하이브리드 광학 NoC에서 전기적인 연결은 웜홀 스위칭(Wormhole switching)과 deterministic 알고리즘인 X-Y 라우팅 알고리즘을 사용하며, 광학적 버스 기반 데이터 전송을 위한 경로 설정 및 광학 라우터 설정을 한다. 광학적 연결에서는 서킷 스위칭(Circuit switching) 방식을 사용하며, 미리 설정된 경로 및 라우터를 이용하여 payload 데이터만 전송을 하게 된다. 그러나 기존에 발표된 하이브리드 광학 NoC같은 경우에는 한 번에 하나의 경로에서만 데이터를 전송 할 수 있다는 단점을 가지고 있어 성능 향상에 한계가 있다. 본 논문에서는 하이브리드 광학 NoC에서 동시에 여러 경로를 이용하여 데이터를 전송하기 위해 전기적인 연결에서 서킷 스위칭 방식과 적응적(adaptive) 알고리즘을 이용하는 새로운 라우팅 알고리즘을 제안하며, 적응적 알고리즘의 문제점인 livelock을 제거할 수 있는 방법 또한 제안한다. 모의실험은 전기적인 NoC, 그리고 웜홀 스위칭 방식의 기존 하이브리드 광학 NoC와 비교 수행 하였다. 그 결과 제안된 방식은 기존 하이브리드 광학 NoC에 비해 60%의 throughput 증가, 그리고 전기적 NoC와 비교했을 때 65%의 전력 감소를 보였다.

Abstract

Networks-on-chip (NoC) is emerging as a key technology to overcome severe bus traffics in ever-increasing complexity of the Multiprocessor systems-on-chip (MPSoC); however traditional electrical interconnection based NoC architecture would be faced with technical limits of bandwidth and power consumptions in the near future. In order to cope with these problems, a hybrid optical NoC architecture which use both electrical interconnects and optical interconnects together, has been widely investigated. In the hybrid optical NoCs, wormhole switching and simple deterministic X-Y routing are used for the electrical interconnections which is responsible for the setup of routing path and optical router to transmit optical data through optical interconnects. Optical NoC uses circuit switching method to send payload data by preset paths and routers. However, conventional hybrid optical NoC has a drawback that concurrent transmissions are not allowed. Therefore, performance improvement is limited. In this paper, we propose a new routing algorithm that uses circuit switching and adaptive algorithm for the electrical interconnections to transmit data using multiple paths simultaneously. We also propose an efficient method to prevent livelock problems. Experimental results show up to 60% throughput improvement compared to a hybrid optical NoC and 65% power reduction compared to an electrical NoC.

Keywords : Hybrid Optical NoC, On-chip routing algorithm, Circuit switching, Adaptive routing algorithm

* 학생회원, ** 평생회원, 성균관대학교 정보통신공학부

(School of Information Communication Engineering, Sungkyunkwan University)

※ 본 연구보고서는 지식경제부 출연금으로 수행한 ETRI 시스템반도체진흥센터 위탁연구의 결과입니다.

접수일자: 2011년4월20일, 수정완료일: 2011년7월18일

I. 서론

최근 몇 년간, 다중 프로세서 시스템-온-칩(MPSoC)의 복잡도가 급격하게 증가하면서 새로운 온-칩 통신구조인 네트워크-온-칩(NoC)이 활발하게 연구되고 있다.^[1] 그러나 기존의 NoC에서는 전기적 상호 연결(Electrical Interconnects, EIs)만을 사용하기 때문에 향후 MPSoC의 복잡도가 더욱 증가되면 성능 향상과 전력 소모의 한계에 직면할 수 있다. 그리하여 기존의 NoC와 광학적 상호 연결(Optical Interconnects, OIs)을 동시에 사용하는 하이브리드 광학 NoC(Hybrid Optical NoC, HONoC)가 추가적으로 연구 되고 있다.^[2]

기존의 HONoC 연구는 임계경로에서만 OIs를 사용하고 임계경로가 아닌 부분에서는 EIs를 사용해 전송 효율을 높이는 방법^[3], VFI(Voltage-Frequency-Island)를 이용해 그룹을 만들어 그룹 간 통신에는 OIs를 사용하고 그룹 내 통신은 EIs를 사용함으로써 지연 시간 및 전력소모를 줄이는 방식^[4] 등이 발표되었다. 또한 대용량의 payload 데이터 전송에는 OIs를 사용하고 크기가 작은 컨트롤 패킷은 EIs를 사용하는 기법^[5], 4개의 코어를 클러스터(cluster)로 묶은 후 클러스터 안에서는 스타 토폴로지와 EIs를 사용하고 클러스터끼리는 메시 토폴로지와 OIs를 사용하는 HOME(Hybrid Optical Mesh NoC)방법^[6]등도 있다.

HONoC에서 사용하는 라우터는 전송 경로가 겹쳐도 라우팅이 가능한 λ -라우터^[7], 크로스바 스위치와 같은 구조를 갖는 광학 크로스바 라우터, EIs와 OIs를 같이 사용하는 하이브리드 라우터^[3~4], EIs와 컨트롤 유닛, 그리고 OIs가 계층적인 구조로 되어있는 HOME 라우터^[6], 광학 도파관(Optical waveguide)과 마이크로 공명기(Micro Resonator, MR)를 가장 적게 사용해서 전력 소모 및 라우터 크기가 작은 Crux 라우터^[8] 등이 있다.

기존의 EIs만 사용하는 NoC나 HONoC는 워홀 스위칭(Wormhole switching)을 근간으로 하는 패킷 스위칭(Packet switching) 방식이 80%이상 사용되며, 라우팅 방법은 90%이상 deterministic 알고리즘을 사용하고 있다.^[9]

[5]에서 제안하는 HONoC의 경우 전기적 네트워크에서는 경로 설정만 하고 광학적 네트워크에서 데이터를 전송한다. 그러나 HONoC의 전기적 네트워크에서 워홀 스위칭과 X-Y 라우팅을 이용해 경로 설정을 하게 되면 여러 경로에서 동시에 데이터를 전송할 경우 현재 사용

중인 광학 라우터의 설정을 변경하거나 광학 라우터가 전송을 마칠 때까지 경로 설정을 못할 수도 있다.

그리하여 본 논문은 HONoC에서 동시에 여러 경로를 이용하여 데이터 전송을 하기 위해 전기적, 광학적 네트워크 모두 서킷 스위칭(Circuit switching) 방식을 사용하며, 적응적 알고리즘을 이용하는 새로운 라우팅 알고리즘을 제안한다. 전기적 네트워크에서 서킷 스위칭을 사용하면 데이터 전송완료 후 경로를 해제해야 한다는 오버헤드가 있으나 본 논문에서는 자동으로 경로를 해제 하는 방법^[6]을 사용하여 이 문제를 해결 하였다. 또한 적응적 알고리즘의 최대 약점인 livelock을 제거할 수 있는 방법도 같이 제안한다.

본 논문의 II절은 제안하는 하이브리드 광학 NoC 구조를 기술하고, III절에서는 라우팅 알고리즘과 프로토콜을 설명하며, IV절에서는 제안하는 구조 및 알고리즘을 사용하여 하이브리드 광학 NoC의 모의실험 결과를 보여준다. 마지막으로 V절에서는 결론을 정리 하였다.

II. 하이브리드 광학 NoC(HONoC) 구조

본 논문에서는 구현의 용이성과 실험의 편의성을 위해 그림 1과 같이 전형적인 4x4 메시 토폴로지 기반으로 설명한다.

HONoC는 광학 라우터, 광학 도파관, 6x6 크로스바 스위치, 그리고 전기적 링크로 이루어져 있으며 광학 라우터는 [8]에서 제안하는 Crux 라우터를 사용하였다. Crux 라우터는 광학 크로스바 라우터나 λ -라우터보다

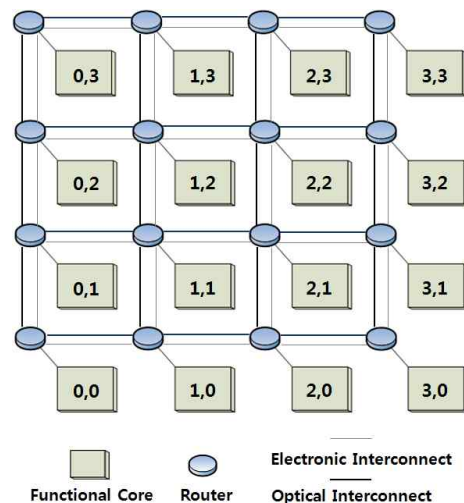


그림 1. 4x4 하이브리드 광학 NoC 구조
Fig. 1. Architecture of a 4x4 Hybrid Optical NoC.

마이크로 공명기와 광학 도파관의 개수가 적어 전력 소모가 적고 사이즈가 작은 장점이 있다.^[7~8]

1. 광학적 네트워크

광학적 네트워크는 광학 라우터와 광학 도파관으로 구성되어 있으며, 광학 도파관은 광학 라우터를 연결하는 역할을 한다. 본 논문에서 광학 라우터는 Crux 라우터(그림 2)^[8]를 사용한다. Crux 라우터는 5x5 non-blocking 라우터이며 동, 서, 남, 북, 그리고 injection/ejection으로 구성된 양방향 포트를 가지고 있다. Crux 라우터 내부에는 그림 3과 같이 두 종류의 1x2 스위칭 요소(Switching element)가 들어가 있다.

각 스위칭 요소들은 하나의 마이크로 공명기(MR)와 두 개의 도파관을 가지고 있으며 공명기의 전원이 켜져 있는 경우와 꺼져 있는 경우에 데이터 전송 방향이 달라진다. 그림 3과 같이 공명기의 전원이 켜져 있으면 데이터 전송은 drop 방향으로 하게 되고 공명기 전원이 꺼져 있으면 데이터 전송은 through 방향으로 하게 된다.

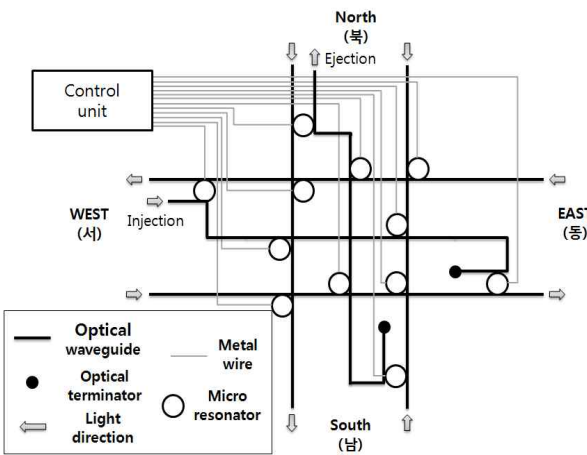


그림 2. Crux 라우터 구조
Fig. 2. Structure of a Crux router.

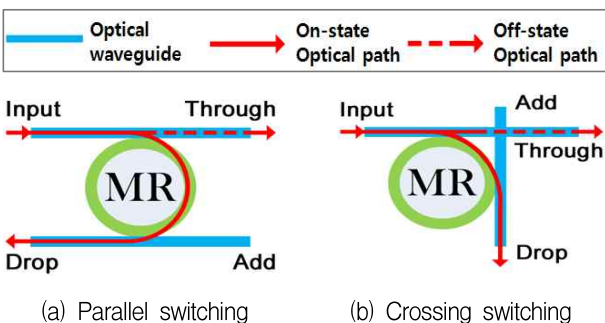


그림 3. 1x2 스위칭 요소
Fig. 3. 1x2 switching element.

하나의 Crux 라우터는 12개의 마이크로 공명기와, 6개의 광학 도파관, 그리고 컨트롤 유닛으로 이루어져 있다. 다른 광학 라우터와 비교했을 때 가장 적은수의 마이크로 공명기와 광학 도파관을 가지고 있으므로 전력 소모와 사이즈에서 이득을 가질 수 있다. 또한 반대방향(동→서, 남→북, injection/ejection)으로 데이터를 전송할 경우 마이크로 공명기의 전원을 켜지 않아도 되므로 전력 소비는 더욱 줄어든다. 컨트롤 유닛은 공명기의 전원을 On/Off 하는 역할을 하며 이는 크로스바 스위치에서 제어 한다.

2. 전기적 네트워크

전기적 네트워크는 16개의 크로스바 스위치와 각 크로스바 스위치를 연결하는 전기적 링크로 구성되어 있다. 각 크로스바 스위치는 동, 서, 남, 북과 Crux 라우터 및 각 코어와 연결된 6개의 양방향 포트로 되어 있다.

전기적 NoC와 관련하여 발표된 대부분의 논문에서는 전송속도가 빠르고 deadlock을 방지 할 수 있는 가상 채널(Virtual channel)을 사용하는 워홀 스위칭과 deterministic 알고리즘으로 복잡도와 latency가 낮은 X-Y 라우팅 알고리즘을 사용한다. 그러나 [5]에서 제안하는 HONoC에서 전기적 네트워크는 payload를 전송하는 것이 아니고 크기가 작은 컨트롤 패킷만 전송하기 때문에 전기적 네트워크에서의 과부하가 적어 워홀 스위칭과 X-Y 라우팅 알고리즘을 사용하지 않아도 된다.

그리하여 본 논문은 전기적 네트워크에서도 서킷 스위칭과 적응적 라우팅 알고리즘을 기반으로 하는 새로운 라우팅 알고리즘과 크로스바 스위치에 벡터 테이블 및 TTL(Time-to-Live) 필드를 추가 하는 방법을 제안해 여러 광학 라우터가 동시에 payload를 전송 할 수 있도록 하였다. 또한 서킷 스위칭의 단점인 경로해제 방식을 자동으로 할 수 있게 하였으며 livelock을 제거 할 수 있는 방법을 제시하였다.

III. 라우팅 알고리즘과 프로토콜

이번 장에서는 본 논문에서 제안하는 알고리즘을 사용하여 HONoC에서 병렬적으로 데이터를 전송하는 방법과 프로토콜에 대하여 설명한다.

1. 라우팅 알고리즘

[5]에서 제안된 HONoC는 그림 4(a)에서 보듯이 (1,

0) 코어에서 (2, 2) 코어로 광학적 네트워크를 이용해 데이터를 전송 중일 때 (0, 0) 코어에서 (3, 3) 코어로 데이터 전송을 위해 경로 설정을 시도하면 (1, 0)과 (2, 0) 코어에 연결된 라우터의 설정이 변경되거나 라우터 사용이 끝날 때 까지 기다려야 하기 때문에 동시 전송이 불가능 하다.

그러나 본 논문에서 제안하는 라우팅 알고리즘을 사용하면 그림 4(b)에서 보는 것과 같이 (1, 0) 코어에서 (2, 2) 코어로 광학적 네트워크에서 데이터를 전송하면

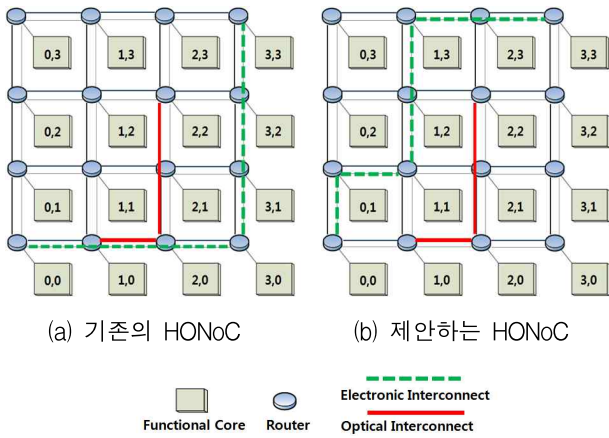


그림 4. 기존의 HONoC와 제안하는 HONoC의 라우팅 알고리즘 비교
Fig. 4. Comparison of routing algorithm between existing and proposed HONoC.

Algorithm 1: Router – X-Dimension

```

Input : Destination address, Source address
1: SET temp ← source
2: IF destination.x > source.x THEN
3:   INCREMENT temp.x
4:   IF temp = block THEN
5:     CALL Y-Dimension
6:   ELSE
7:     RETURN Next router (source.x+1, source.y)
8:   END IF
9: ELSE IF destination.x < source.x THEN
10:  DECREMENT temp.x
11:  IF temp = block THEN
12:    CALL Y-Dimension
13:  ELSE
14:    RETURN Next router (source.x-1, source.y)
15:  END IF
16: ELSE
17:  CALL Y-Dimension
18: END IF

```

OUTPUT: NEXTROUTER ADDRESS

그림 5. X-Dimension 라우팅 알고리즘
Fig. 5. Routing algorithm for X-Dimension.

서 (0, 0) 코어에서 (3, 3) 코어까지 전기적인 네트워크에서 새로운 경로를 설정해 통신할 수 있다. 제안하는 알고리즘은 서킷 스위칭을 사용하여 한번 경로가 설정되면 다른 컨트롤 패킷의 접근을 방지한다.

서킷 스위칭을 사용하면서 기존에 연구된 HONoC의

Algorithm 2: Router – Y-Dimension

```

Input : Destination address, Source address
1: SET temp ← source
2: IF destination.y > source.y THEN
3:   INCREMENT temp.y
4:   IF temp = block THEN
5:     SET temp.y ← source.y-1
6:     IF temp = block THEN
7:       RETURN rollback
8:     ELSE
9:       RETURN Next router (source.x, source.y-1)
10:    END IF
11:  ELSE
12:    RETURN Next router (source.x, source.y+1)
13:  END IF
14: ELSE IF destination.y < source.y THEN
15:  DECREMENT temp.y
16:  IF temp = block THEN
17:    SET temp.y ← source.y+1
18:    IF temp = block THEN
19:      RETURN rollback
20:    ELSE
21:      RETURN Next router (source.x, source.y+1)
22:    END IF
23:  ELSE
24:    RETURN Next router (source.x, source.y-1)
25:  END IF
26: ELSE
27:  IF Prerouter.y > source.y THEN
28:    DECREMENT temp.y
29:    IF temp = block THEN
30:      RETURN rollback
31:    ELSE
31:      RETURN Next router (source.x, source.y-1)
32:    END IF
33:  ELSE IF Prerouter.y < source.y THEN
34:    INCREMENT temp.y
35:    IF temp = block THEN
36:      RETURN rollback
37:    ELSE
38:      RETURN Next router (source.x, source.y+1)
39:    END IF
40:  END IF
41: END IF

```

OUTPUT: NEXTROUTER ADDRESS

그림 6. Y-Dimension 라우팅 알고리즘
Fig. 6. Routing algorithm for Y-Dimension.

deterministic 알고리즘을 사용하게 되면 (1, 0) 코어처럼 미리 설정된 광학적 네트워크 경로가 데이터를 전송하고 있을 때 데이터 전송이 완료되고 경로가 해제되기 전까지 다른 전기적 네트워크의 패킷이 경로를 설정하지 못하고 기다리게 되어 지연이 발생한다. 그리하여 본 논문에서는 서킷 스위칭과 적응적 알고리즘을 기반으로 하는 새로운 라우팅 알고리즘을 적용하였다.

제안하는 라우팅 알고리즘의 의사코드(Pseudocode)는 그림 5, 6과 같고 순서도는 그림 7과 같다. 패킷이 라우터에 도착하면 해당 코어의 X축 좌표와 목적지 X축 좌표를 비교해서 목적지 X축 좌표가 더 크면 X값에 +1, 목적지 X축 좌표가 더 작으면 X값에 -1을 한 후 벡터 테이블의 포트를 확인한다. 이때 벡터 테이블에서 해당 포트가 block이나 rollback로 되어 있으면 Y-Dimension을 호출한다.

Y-Dimension에서도 코어의 Y축 좌표와 목적지 Y축 좌표를 비교해서 크면 +1, 작으면 -1을 한 후 벡터 테이블에서 포트를 확인한다. 만약 포트가 블로킹(Blocking) 되어 있으면 반대방향 포트를 확인 한다. 벡터 테이블에서 반대 방향 포트도 블로킹 되어 있으면 바로 앞서 패킷을 전송했던 라우터에게 rollback 패킷을 전송한다. 만약 중간에 블로킹 되어있는 포트가 없어서 해당 포트로 패킷을 전달하게 되면 패킷을 전달한 라우터는 자신이 사용 중 이라는 패킷을 자신과 연결된 라우터에 알려 벡터 테이블을 갱신한다.

X-Dimension에서 코어의 X 좌표와 목적지 X 좌표가 같으면 Y-Dimension을 바로 호출한다. 코어 X 좌표와 목적지 X 좌표가 다른 경우에 Y-Dimension값이 같

으면 이전 라우터와 비교를 해서 처리 하며, 코어 X 좌표와 목적지 X 좌표가 같은 경우에 Y-Dimension 값이 같으면 해당 코어가 패킷의 목적지가 된다.

그림 4 (b)를 보면 (1, 0)에서 (2, 2)까지 광학적 네트워크에서 데이터 전송이 진행 중 일 때 제안하는 라우팅 알고리즘을 이용해 (0, 0)에서 (3, 3)까지의 경로 설정을 동시에 할 수 있다. (0, 0)에서 패킷이 생성되면 X값이 증가하는 (1, 0) 라우터를 확인 후 (1, 0) 라우터가 동작 중이면 Y값을 증가 시켜 (0, 1) 라우터를 확인한다. (0, 1) 라우터가 동작하지 않고 있으면 (0, 1) 라우터로 전송한다. 그 후 X 축인 (1, 1) 라우터로 패킷을 전송 후 (2, 1), (2, 2) 코어가 사용 중 이기 때문에 X축으로 진행하지 못하고 Y축으로 진행해 (1, 3)까지 전송된다. (1, 3) 라우터부터 (3, 3) 라우터 까지 중간에 사용하는 라우터가 없기 때문에 X값만 증가 시켜 전송을 하게 된다.

위와 같이 본 논문에서 제안하는 알고리즘은 X-Y 라우팅 알고리즘보다 latency가 약간 증가하나 여러 경로에서 병렬적으로 통신이 가능하기 때문에 그에 비해 더욱 높은 throughput 을 보인다.

2. 프로토콜

본 논문에서 컨트롤 패킷은 출발지 주소, 목적지 주소, 이전 라우터 주소, 지나온 홉(Hop) 개수, TTL 필드, 패킷 사이즈로 구성이 된다.

새로운 라우팅 알고리즘에서 사용되는 전송 프로토콜은 광학 라우터 설정, 상태 전송, Ack 전송, 데이터 전송, 광학 및 전기적 네트워크 경로 해제로 총 5단계로 되어 있으며 위의 프로토콜을 이용해 HONoC에서 네트워크 전송 효율을 증가 시킬 수 있다.

광학 라우터 설정은 각 라우터가 컨트롤 패킷을 확인 후 데이터 전송 경로에 따라 광학 라우터의 컨트롤 유닛을 제어 하여 마이크로 공명기를 설정 하는 것이다.

상태 전송은 현재 크로스바 스위치에서 다른 크로스바 스위치로 패킷을 전송 하거나 현재 크로스바 스위치가 패킷의 목적지 크로스바 스위치면 현재 크로스바 스위치와 연결된 모든 링크에 블로킹 신호를 보내 인접한 스위치의 벡터 테이블을 갱신한다.

Ack 및 데이터 전송은 데이터를 전송하기 위한 경로가 설정 되면 설정된 광학적 네트워크를 이용해 Ack를 전송하며 Ack를 받은 출발지 코어 역시 설정되어진 광학적 네트워크 경로를 이용해 데이터를 전송한다.

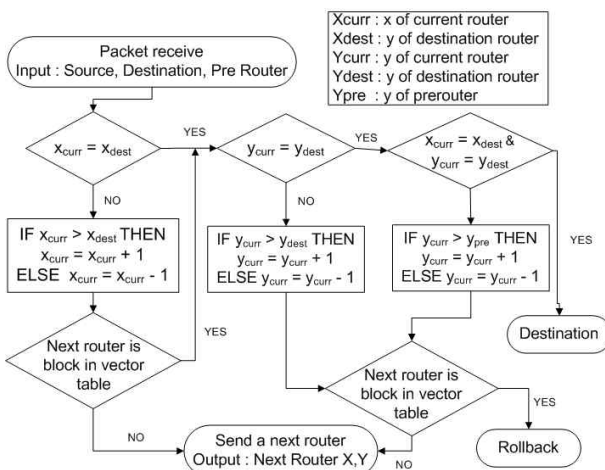


그림 7. 제안하는 라우팅 알고리즘 순서도
Fig. 7. Flowchart of proposed routing algorithm.

데이터 전송을 완료하면 광학 및 전기적 네트워크 모두 서킷 스위칭을 사용하기 때문에 설정된 경로를 해제시켜야 된다. 기존 HONoC의 경우에는 데이터 전송이 완료 후 해제 패킷을 전송하는 반면 [6]에 나와 있는 해제 방법을 사용하면 성능을 향상 시킬 수 있다.

이 해제 방식은 Ack를 통해 받은 전송경로의 라우터 정보를 이용해 식(1)과 같이 C_{tx} 값을 구해 해제 패킷의 TTL값에 넣어 데이터 전송 시작과 동시에 해제 패킷을 전송 하도록 한다. 이때 각 크로스바 스위치는 TTL 값을 저장하고 있다가 각 클럭마다 값을 감소시켜 TTL 값이 0이 되면 전송이 완료 됐다고 판단하고 해당 경로를 해제 시킨다. 해제 패킷이 다음 스위치로 전송 되면 전송할 때 걸리는 클럭 사이클 수만큼 TTL 값이 감소되어 스위치에 저장되며, 목적지 까지 전송되지 않고 TTL 값이 0이 되면 즉시 전송경로를 해제 한다.

$$C_{tx} = \left\lceil \frac{(P_{size} \cdot f_{elec})}{B_{optical}} \right\rceil \quad (1)$$

식 (1)에서 C_{tx} 는 데이터를 전송할 때 걸리는 클럭 사이클 이며 P_{size} 는 전송하는 패킷 크기이며, f_{elec} 은 전기적 네트워크 클럭 주파수이고 $B_{optical}$ 은 광학적 네트워크 대역폭 이다.

서킷 스위칭과 적응적 라우팅 알고리즘을 사용할 때의 문제점으로는 livelock에 취약하다는 점이 있다. 그리하여 livelock을 제거 할 수 있는 방법도 추가 제안하였다.

그림 8에서 보듯이 (0, 2) 코어에서 (3, 2)코어로 데

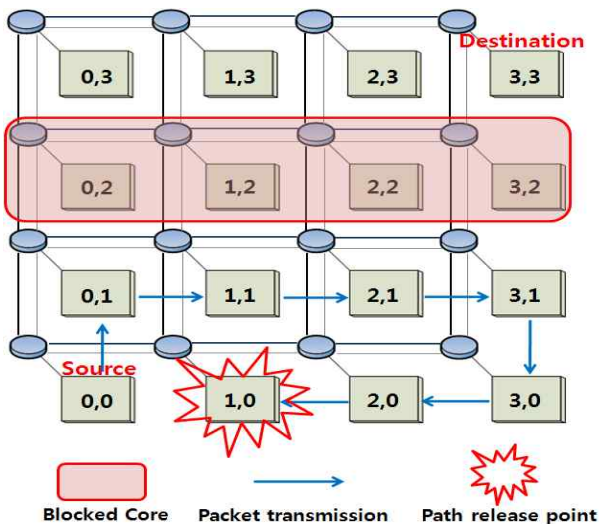


그림 8. 새로운 라우팅 알고리즘에서 livelock 상황
Fig. 8. Case of lovelock in new routing algorithm.

이터가 전송 중 일 때 (0, 0)에서 (3, 3)까지 경로를 설정을 시도 하면 Y축 2번 코어들이 전송 완료되기 전까지 livelock이 생기게 된다. 이러한 livelock을 방지하기 위해 식 (2)와같이 C_{cnt} 값을 구해 컨트롤 패킷을 전송할 때 TTL 필드에 저장시켜 전송한다. 컨트롤 패킷의 TTL 값 역시 해제 패킷과 같이 크로스바 스위치에 저장되며 각 사이클 마다 TTL 값을 감소시키며 각 홉을 넘어갈 때는 해당 클럭만큼 감소시킨다. TTL 값이 0이 되면 현재 설정된 경로를 해제 하고 해당 경로 때문에 전송을 하지 못하고 있던 다른 코어간의 전송을 할 수 있게 함으로서 livelock을 제거 한다.

$$C_{cnt} = \left\lceil \left(\frac{P_{size} \cdot f_{elec}}{B_{elec}} \cdot \frac{h}{2} \right) + S_{aps} \frac{h}{2} \right\rceil \quad (2)$$

수식 (2)에서 C_{cnt} 은 컨트롤 패킷이 전기적 네트워크에서 유지될 수 있는 총 사이클 수이다. 또한 P_{size} 는 컨트롤 패킷 사이즈 이며, f_{elec} 은 전기적 네트워크 클럭 주파수 이고 B_{elec} 은 전기적 네트워크의 대역폭이다. S_{aps} 는 크로스바 스위치가 동작하는데 필요한 평균 사이클 수이며 h 는 칩 안에 있는 총 스위치 개수이다.

IV. 실험 및 결과

본 논문에서는 각 라우터 및 코어의 병렬적 처리, 그리고 클럭 기반의 타이밍 등 구현의 용이성 위해 SystemC로 작성한 시뮬레이터를 사용하였다.

실험은 EIs만 사용하는 NoC와 병렬 전송을 하지 못하는 HONoC, 그리고 본 논문에서 제안하는 HONoC를 이용해 10MByte 만큼의 데이터를 무작위로 전송 한 후 그에 대한 throughput과 전력 소비를 비교 하였다.

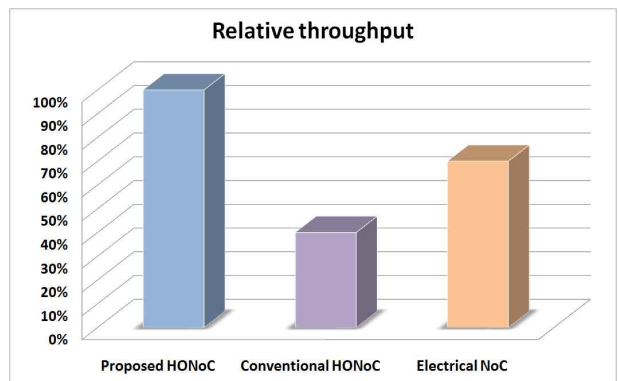


그림 9. 각 알고리즘에서의 상대적 throughput
Fig. 9. Relative throughput in each algorithm.

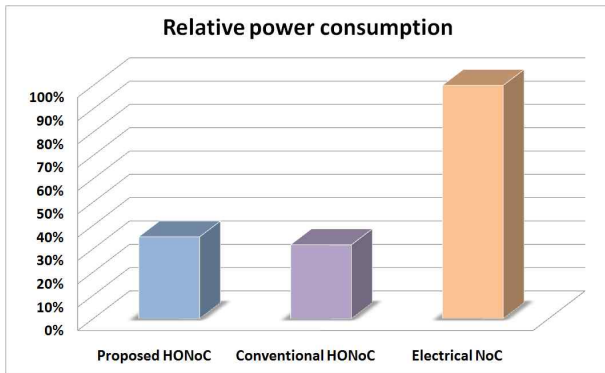


그림 10. 각 알고리즘에서의 상대적 전력소모
Fig. 10. Relative power consumption in each algorithm.

그림 9는 throughput에 대한 결과를 보여준다. 병렬 전송을 못하는 HONoC의 throughput이 가장 낮은 이유는 서로 이웃한 코어끼리 데이터를 전송하게 되면 HONoC의 경우 전기신호를 광신호로 변경하고 다시 전기신호로 변경하는데 걸리는 시간 때문에 EIs만 사용하는 NoC보다 50%가량 낮은 throughput을 갖기 때문이다. 그러나 멀리 떨어진 코어끼리 데이터를 전송하게 되면 HONoC가 38%가량 높은 throughput을 갖는다. 이러한 결과로 병렬 전송을 못하는 HONoC는 EIs만 사용하지만 병렬 전송이 가능한 NoC 보다 30%가량 낮은 throughput를 보이고 본 논문에서 제안하는 HONoC는 60%가량 높은 throughput를 보인다.

그림 10은 각 경로에서 데이터를 전송하는 것에 대한 전력 소비를 보여준다. 전력 소비는 기본적으로 모든 HONoC가 EIs를 사용하는 NoC보다 적은 전력 소비를 한다. 그러나 제안하는 HONoC가 병렬 전송을 못하는 HONoC보다 전력 소모가 많은 이유는 적응적 알고리즘의 단점인 livelock이 생기는 문제와 더 이상 경로 설정을 못할 시 rollback을 하기 때문이다. 비록 병렬 전송을 못하는 HONoC보다 4%의 전력을 더 사용하지만 throughput이 60%가량 증가하기 때문에 효율성면에서는 더욱 좋다고 생각된다. 또한 EIs만을 사용하는 NoC보다 제안하는 HONoC가 65%가량의 적은 전력 소비를 할 수 있다.

V. 결 론

본 논문은 HONoC에서 병렬적으로 데이터를 전송하기 위해 전기적 네트워크에서 서킷 스위칭 방식과 적응적 알고리즘을 이용하는 새로운 라우팅 알고리즘을 제

안 하였다. 또한, 적응적 알고리즘의 약점인 livelock을 제거 할 수 있는 방법도 함께 제안하였다. 서킷 스위칭에서 경로 해제 및 livelock제거를 자동으로 하면서 병렬적으로 데이터를 전송하기 때문 기존의 NoC 및 HONoC보다 더욱 높은 throughput 및 낮은 전력 소비를 보일 수 있었다.

제안하는 HONoC는 기존의 병렬 전송이 어려운 HONoC보다 4%의 전력 더 사용 하지만 60%가량 throughput 증가를 보인다. 또한 EIs만을 사용하는 NoC보다는 throughput 증가 및 전력 소비가 감소하는 현상을 볼 수 있었다.

본 논문에서는 구현의 복잡성 때문에 HONoC의 구조를 4x4로 한정해서 실험을 하였다. 그러나 코어가 많아질수록 더욱 높은 성능 향상이 예상됨으로 향후에 3D-IC 기술을 접목시킨 HONoC에 관하여 연구를 진행하면 더욱 좋은 성능을 기대 할 수 있을 것이다.

참 고 문 헌

- [1] L. Benini and G. D. Micheli, "Network on chip: A new SoC paradigm," *IEEE Computer*, pp.70-78, January, 2002.
- [2] Raymond G. Beausoleil, Philip J. Kuekes, Gregory S. Snider, Shih-Yuan Wang, R. Stanley Williams, "Nanoelectronic and Nanophotonic Interconnect", *Proceeding of the IEEE*, pp.230-247, February, 2008.
- [3] 권순태, 조준동, 한태희, "광학적 상호연결을 이용한 네트워크-온-칩에서의 스위치 구조와 라우팅 최적화 방법", *대한전자공학회 논문지*, 제46권 SD편, 제9호, 25-32쪽, 2009년 9월
- [4] 서정택, 황용중, 한태희, "전압-주파수-구역과 광학적 상호 연결을 이용한 네트워크-온-칩에서 통신 최적화 기법", *대한전자공학회 추계 학술대회 논문집*, 제33권 2호, 28-29쪽, 2010년 11월
- [5] H. Gu, K. H. Mo, J. Xu, W. Zhang, "A Low-Power Low-Cost Optical Router for Optical Networks-on-Chip in Multiprocessor Systems-on-Chip", *VLSI, 2009. Proceeding. IEEE Computer Society Annual Symposium on*, pp.19-24, May, 2009.
- [6] K. H. Mo, Y. Ye, X. Wu, W. Zhang, W. Liu, J. Xu, "A Hierarchical Hybrid Optical-Electronic Network-on-Chip", *VLSI, 2010 IEEE Computer Society Annual Symposium on*, pp.327-332, July, 2010.
- [7] M. Briere, B. Girodias, et al, "System Level

Assessment of an Optical NoC in MPSoC Platform”, Design, Automation & Test in Europe Conference & Exhibition, pp.1084-1089, April, 2007.

[8] Y. Xie, M. Nikdast, J. Xu, et al, “Crosstalk Noise and Bit Error Rate Analysis for Optical Network-on-Chip”, Design, Automation Conference, 2010 47th ACM/IEEE, pp.657-660, June, 2010

[9] E. Salminen, A. Kulmala and T. D. Hamalainen, “Survey of Network-on-Chip Proposals”, White Paper OCP-IP, 2008.

— 저 자 소 개 —



서 정 택(학생회원)
 2010년 한림대학교 컴퓨터공학과
 학사 졸업.
 2010년 3월~현재 성균관대학교
 정보통신공학부 석사과정.

<주관심분야 : 임베디드 시스템, SoC 설계>



황 용 중(학생회원)
 2010년 성균관대학교 전자전기
 공학과 학사 졸업.
 2010년 3월~현재 성균관대학교
 정보통신공학부 석사과정.

<주관심분야 : 임베디드 시스템, SoC 설계>



한 태 희(평생회원)-교신저자
 1992년 KAIST 전기 및
 전자공학과 학사.
 1994년 KAIST 전기 및
 전자공학과 석사.
 1999년. KAIST 전기 및
 전자공학과 박사.

1999년 3월~2006년 8월 삼성 전자 통신연구소
책임 연구원.

2006년 9월~2008년 2월 한국산업기술대학교
전자공학과 조교수.

2008년 3월~현재 성균관대학교 정보통신공학부
반도체시스템공학 전공 부교수.

<주관심분야 : IT SoC 설계 및 설계 방법론,
단말 시스템, IT 융합 기술>