

논문 2009-46SD-11-8

# 지연시간 효율 개선을 위한 On-Chip Network 구조 설계 및 구현

( Design and Implementation of On-Chip Network Architecture for  
Improving Latency Efficiency )

조성민\*, 조한욱\*, 하진석\*, 송용호\*\*

( Seongmin Jo, Han Wook Cho, Jin Seok Ha, and Yong Ho Song )

## 요약

최근 SoC의 집적도가 증가함에 따라 칩 내부의 통신 효율성은 시스템 성능에 직접적인 영향을 미치고 있다. 이에 따라 칩 내부의 통신 메커니즘은 과거 shared wire를 이용한 버스 시스템에서 라우터를 기반으로 하는 NoC로 진화하고 있다. 하지만, NoC 내부의 라우터는 컨트롤 로직이 복잡해짐에 따라 신호 전달 과정에서 지연시간을 증가시켜 NoC의 성능을 제한시킨다. 따라서 본 논문에서는 이러한 지연시간을 개선시키기 위하여 낮은 복잡도를 갖는 라우터를 제시한다. 제안한 라우터의 구조 검증 및 성능 평가를 위하여 ESL 기법의 시뮬레이션 플랫폼을 구축하였다. 본 논문에서 제안한 NoC 구조는 기존의 VC 라우터 기반의 NoC에 비해 대역폭은 약 1-2% 정도 감소하였지만, 평균적으로 약 50%의 지연시간이 감소 효과를 보였다.

## Abstract

As increasing the number of IPs integrated in a single chip and requiring high communication bandwidth on a chip, the trend of SoC communication architecture is changed from bus- or crossbar-based architecture to packet switched network architecture, NoC. However, highly complex control logics in routers require multiple cycles to switch packet. In this paper, we design low complex router to improve the communication latency. Our NoC design is verified by simulation platform modeled by ESL tool, SoC Designer. We also evaluate our NoC design comparing to the previous NoC architecture based on VC router. Our results show that our NoC architecture has less communication latency, even small throughput degradation (about 1-2%).

**Keywords :** SoC, NoC, ESL, low latency, on-chip network

## I. 서론

반도체 공정 기술의 발전으로 인해서 칩에 집적되는 IP 코어 수가 급격하게 증가하고 있다. 이와 함께 IP 코

어 간의 통신 요구량이 증가하였고, 기존의 버스 및 크로스바 기반의 통신 구조로는 이러한 통신 요구량 수용에 어려움이 있었고, 이를 대체할 새로운 통신 구조가 필요하게 되었다. 따라서 최근 네트워크 구조를 이용한 Network-on-Chip (NoC)에 대한 연구가 활발히 진행되고 있다.

\* 학생회원, \*\* 정회원, 한양대학교 전자컴퓨터통신공학과 (Department of Electronics and Computer Engineering, Hanyang University)

※ 본 연구보고서는 지식경제부 출연금으로 ETRI, 시스템반도체산업진흥센터에서 수행한 ITSoc 핵심설계인력양성산업의 연구결과입니다.

※ 본 연구는 2009년도 「서울시 산학연 협력사업」의 「나노 IP/SoC 설계기술 혁신 사업단」의 지원으로 이루어졌습니다.

접수일자: 2009년9월14일, 수정완료일: 2009년10월24일

NoC는 패킷을 전달하는 라우터와 IP 코어를 연결하는 네트워크 인터페이스로 구성된 네트워크 시스템이다. NoC는 멀티홉 패킷 전달 시스템을 구현하여 동시에 다수개의 패킷을 전달할 수 있기 때문에 높은 통신 대역폭을 제공한다. 뿐만 아니라 많은 수의 IP 코어를 효율적으로 지원 가능하기 때문에 높은 확장성을 제공

한다<sup>[1]</sup>.

하지만 NoC에서는 멀티홉 전송 및 IP 코어와 네트워크 간의 프로토콜 변환 과정으로 인해 기존의 버스 및 크로스바 기반의 통신 구조에 비해 높은 지연시간을 가진다. 통신 채널의 접근에 대하여 경합이 없는 경우에 버스 및 크로스바 기반의 통신 구조에서는 단일 사이클에 IP 코어 간의 통신이 가능하다. 반면에, NoC 에서의 IP 코어 간 통신 시간은 IP 코어 사이에 존재하는 라우터의 수, 라우터 파이프라인, 네트워크 인터페이스의 프로토콜 변환 지연시간 등에 의해 결정된다. 따라서 NoC에서는 통신 채널의 접근에 대한 경합이 없더라도 IP 코어 간의 통신은 다수 사이클의 지연시간의 발생한다. 이러한 통신 지연시간 상승은 프로세서와 같은 특정 IP 코어에 대해서는 성능에 큰 영향을 끼칠 수 있다.

라우터 파이프라인 구조를 최적화함으로써 NoC의 지연시간을 단축시키는 것이 가능하다. 라우터 내부에 구현되는 Virtual Channel (VC) 수가 증가함에 따라 Virtual channel Arbitration (VA)과 Switch Arbitration (SA) 로직의 복잡도가 높아지게 되며, 내부 동작에 필요한 파이프라인 단계가 많아지게 되어 지연시간을 증가시키게 된다.

본 논문에서는 기존 라우터의 구조를 최적화한 Scalable On-chip network for Large Application (SOLA) NoC 구조를 제안한다. 제안하는 SOLA NoC 구조는 다음과 같은 특징을 갖고 있다.

- Pre-Routing Computation (PRC) 로직은 패킷이 통과하려는 라우터의 출력 포트 정보를 미리 계산하여 입력 패킷에 대한 라우팅 작업이 이루어지는 시간에 VA/SA 작업을 동시 수행할 수 있게 되어 라우터 내부 동작에 대한 지연 시간을 개선한다.
- 입력 패킷은 PRC에 의해 계산된 출력 포트에 마련된 버퍼에 분류가 되기 때문에 VA와 SA 내부의 스케줄링에 필요한 중재기의 수가 적어지고, 각 중재기 별로 요청 승인 신호선의 수가 줄어든다. 따라서 제안된 라우터의 SA/VA에 대한 설계 복잡도를 낮출 수 있어 단일 사이클 내에 VA와 SA가 동시에 동작 가능하다.

제안한 NoC 구조에 대한 성능 검증 및 평가를 위해 Electronic System Level (ESL)<sup>[2]</sup> 모델링 기법을 이용

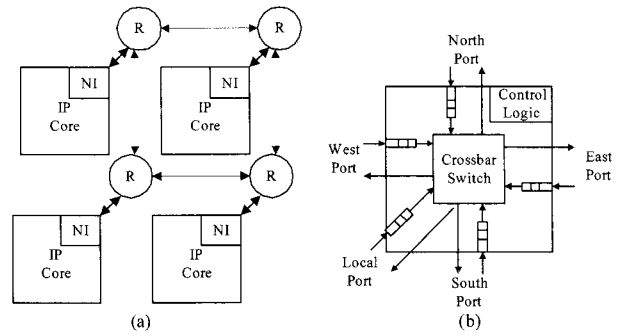


그림 1. (a) NoC 및 (b) 라우터 구조  
Fig. 1. The architectures of (a) NoC and (b) router.

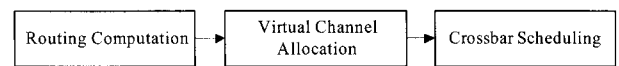


그림 2. VC 라우터의 3단계 파이프라인 구조  
Fig. 2. The 3-stage pipeline architecture of VC router.

하였다. ESL 디자인 모델을 이용함으로써 제안된 시스템의 하드웨어와 소프트웨어 요소에 대한 높은 추상화 단계에서 성능 평가가 가능하다. 또한 ARM11용 모델을 사용할 수 있으므로, 제안한 NoC를 기반으로 하는 전체 시스템 모델링이 가능하다.

본 논문의 구성은 다음과 같다. II장에서는 NoC 통신 구조에 대하여 간략하게 설명하고 III장에서는 NoC의 지연시간을 최적화하기 위한 기존 연구를 소개한다. IV장에서는 본 논문에서 제안하는 SOLA NoC 구조에 대해서 설명하고, V장에서는 시뮬레이션 플랫폼을 이용하여 SOLA의 기능 검증 및 성능 평가 결과를 제시하고 VI장에서 결론으로 논문을 마무리 짓는다.

## II. NoC 통신 구조

확장성을 가진 SoC 통신 구조로서 패킷 스위치 기반의 통신을 지원하는 NoC 구조에 대하여 많은 연구가 진행되고 있다<sup>[1, 3-6]</sup>. 그림 1(a)는 NoC의 일반적 구조를 보이고 있다. 내부에는 IP와 NoC를 연결하는 네트워크 인터페이스와 네트워크 내부에서 목적지를 향해 패킷을 전달하는 라우터로 구성된다. 따라서 NoC는 다수 개의 라우터의 연결로 구성되어 있기 때문에 패킷이 경로에 존재하는 라우터를 거쳐서 목적지까지 전송되는 멀티홉 통신 구조를 갖고 있다.

NoC는 패킷 스위치 기반의 통신을 사용하기 때문에 기존 IP의 신호를 패킷 형태로 변환하여 전송해야 한다. 이렇게 함으로써 NoC의 라우터 간 신호선 수는 IP의 입출력 선

호선 수에 비해 작아질 수 있다.

전형적인 VC 라우터<sup>[5]</sup>의 패킷 전송과정은 3 단계로 구성된다. 먼저 라우팅 로직은 라우터로 들어온 패킷에 대해 출력 포트를 선정한다. 다음으로 VA 로직은 출력 포트에 구현된 여러 개의 가상 채널 중 하나를 선정한다. 마지막으로 SA 로직은 크로스바 스위치에서 같은 출력 포트에 전달하고자 하는 패킷 중 하나를 선정한다. 하지만 라우터 컨트롤 로직은 3단계의 작업을 순서대로 진행하여야 하며, SA 및 VA의 복잡도가 높기 때문에 그림 2와 같이 파이프라인 형태로 동작한다\*<sup>[5]</sup>.

네트워크 인터페이스는 IP 코어에서 전송하고자 하는 신호선 정보를 패킷 형태로 변환한다. 또한 통신하려는 두 IP 코어 사이에 종단 간 통신을 제공한다. 따라서 일반적으로 네트워크 인터페이스는 ISO/OSI 참조 모델<sup>[7]</sup>과 같은 계층 구조를 사용한다.

### III. 관련 연구

NoC는 높은 대역폭 제공하고 확장성을 제공함에도 불구하고 멀티홉 통신 구조 및 프로토콜 변환으로 인해 높은 지연시간이 발생한다. 따라서 NoC 지연시간을 최소화하기 위하여 라우터 구조를 최적화하는 연구 진행되고 있다.

라우터에서 발생하는 지연시간을 감소시키기 위하여 Speculative VC 라우터<sup>[5]</sup>는 VA와 동시에 SA를 수행하여 다음에 할당할 스위치 접근 권한을 미리 예측하는 방법을 사용한다. SA가 정확하게 스위치 접근 권한을 예측하는 경우 VA와 SA를 동시에 수행하는 효과를 통하여 1 사이클의 지연시간을 줄일 수 있다. 하지만 예측이 맞지 않는 경우 기존의 VC 라우터와 같은 지연시간을 가진다. 하지만 Speculative VC 라우터는 라우터 구조를 최적화한 것이 아니라 Speculation 로직을 추가하는 방법을 사용하였다.

라우터의 지연시간에 가장 큰 영향을 주는 VA와 SA 복잡도는 라우터의 입력 포트에 존재하는 VC 수에 따라 급격하게 증가한다. 그 원인은 각 VC에 있는 패킷은 어떠한 출력 포트에 대해서 선택이 가능하고, 선택된 출력 포트와 연결된 어떠한 VC에 대해서 선택이 가능하기 때문이다. 따라서 VA와 SA는 각 입력 VC마다 모든 경우의 수에 대해서 중재하여야 하기 때문에 복잡

\* 패킷은 SA를 통해 스위치 통과 허락을 받는 즉시 스위치 통과가 가능하다고 가정한다.

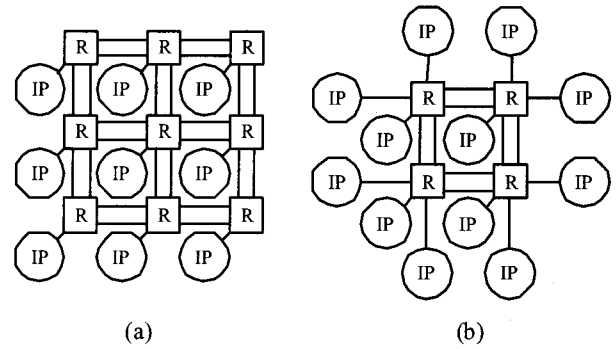


그림 3. SOLA 구조에서 지원 가능한 네트워크 topology, (a) mesh topology, (b) application-specific topology

Fig. 3. The network topology supported by SOLA NoC, (a) mesh topology, (b) application-specific topology.

도가 급격하게 증가하는 것이다. 본 논문에서는 이러한 VA와 SA의 높은 복잡도를 감소시키기 위하여 각 VC는 특정한 출력 포트에 전송 가능하도록 하여 라우터 구조를 최적화하는 방법에 대해서 제시한다.

## IV. SOLA 구조 설계

### 1. SOLA 구조

SOLA는 NoC 기반의 SoC 통신 구조로서, 5개의 입력 포트와 출력포트와 wormhole 패킷 스위칭<sup>[9]</sup>를 구현한 라우터와 다양한 IP 통신 프로토콜을 지원할 수 있는 네트워크 인터페이스로 구성된다. 라우터의 연결을 통하여 그림 3과 같이 2D mesh 구조를 구성할 수 있을 뿐만 아니라, application-specific topology도 구성할 수 있다. 2D mesh 구조에서는 XY 라우팅 기법<sup>[9]</sup>를 이용하지만, application-specific topology는 XY 라우팅 기법을 이용하는 경우에 패킷이 막다른 경로로 진행될 수 있기 때문에 라우팅 테이블을 이용하여 패킷을 전달한다.

### 2. 디자인 고려 사항

#### 1) 지연시간 발생 요소

NoC는 멀티홉 패킷 통신 구조이기 때문에 기존 SoC 통신 구조에 비해 긴 지연시간을 가진다. 전체 패킷 통신 지연시간( $T_{packet}$ )은 수식 (1)과 같이 통신하려는 두 IP 코어 사이에 있는 라우터에서 발생하는 지연시간

$$\left( \sum_{n=1}^h T_{router(n)} \right) \text{과 두 IP 코어와 라우터로 구성된 네트}$$

워크 사이에 존재하는 네트워크 인터페이스에서 패킷 프로토콜과 IP 코어가 사용하는 프로토콜 사이의 변환으로 인하여 발생하는 지연시간 ( $T_{ni\_src} + T_{ni\_dst}$ )의 합으로 나타낼 수 있다.

$$T_{packet} = T_{ni\_src} + \sum_{n=1}^h T_{router(n)} + T_{ni\_dst} \quad (1)$$

라우터 및 네트워크 인터페이스의 지연시간은 수식 (2), (3)와 같이 네트워크 혼잡이 없더라도 필연적으로 나타나는 지연시간( $T_{router\_pipe}$ ,  $T_{ni\_layer}$ )과 네트워크에서 채널 접근 경쟁으로 인해 발생하는 지연시간( $T_{router\_contention}$ ,  $T_{ni\_contention}$ )으로 분류할 수 있다.

$$T_{router(n)} = T_{router\_pipe} + T_{router\_contention} \quad (2)$$

$$T_{ni} = T_{ni\_layer} + T_{ni\_contention} \quad (3)$$

NoC의 지연시간 개선을 위해서는 수식 (2), (3)의 두 지연시간 요소의 감소가 필요하다. 첫 번째 지연 시간 요소를 감소시키기 위해서는 라우터 구조의 최적화를 통해서 최대한 두 IP 코어 간의 통신에 발생하는 불필요한 지연시간 요소를 제거하여야 한다. 두 번째 지연 시간 요소를 감소시키기 위해서는 네트워크의 혼잡 상황을 제어하여 서로 다른 패킷 흐름 간의 충돌을 최소화하는 것이 필요하다. 두 지연 시간에 대해서 최소화하는 것은 고성능 NoC를 디자인을 위해서 중요한 요소이지만, 본 논문에서는 첫 번째 지연시간 요소를 감소시키는 것을 목표로 한다.

### 2) 라우터의 디자인 고려 사항

라우터에서 발생하는 지연시간 발생의 큰 요소는 컨트롤 로직 중 VA와 SA이다. VA는 서로 다른 입력 버퍼에 있는 패킷에 대해서 경로 상 다음 라우터의 입력 버퍼를 선택한다. VC 라우터에서는 VA를 그림 4(a)와 같이 2단계의 중재 로직으로 구현한다<sup>[5]</sup>. 첫 단계에서 각 입력 포트에서 같은 출력 포트로 전송되는 패킷의 가상 채널 할당 요청에 대하여 중재를 수행한다. 첫 단계에서 각 입력 포트 중재를 통해 권한을 얻은 요청은 두 번째 단계에서 같은 가상 채널 요청에 대하여 중재를 수행한다. 두 번째 단계의 중재를 통해 권한을 얻은 가상 채널 요청은 성공적으로 가상 채널을 할당받고 해당 패킷은 SA 단계로 진행된다.

VC 라우터에서 VA는 2 단계의 중재 과정이 필요하

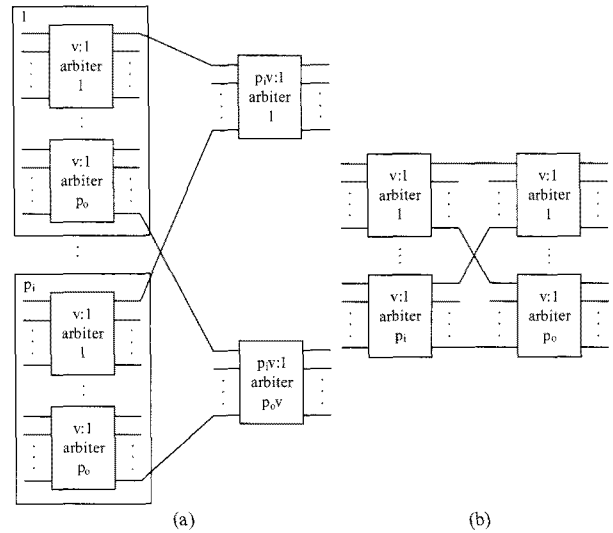


그림 4. VC 라우터의 VA 및 SA 구조[5], (a) VA, (b) SA  
Fig. 4. The VA and SA architectures of VC router, (a) VA, (b) SA.

고 많은 수의 중재기가 필요하기 때문에 구현 복잡도가 높다. 입력 포트의 수를  $P_i$ , 출력 포트의 수를  $P_o$ , 가상 채널의 수를  $v$ 라고 하면 VA에서 필요한 총 중재기 개수는 수식 (4)로 표현할 수 있다. 5 포트의 입출력 포트와 5개의 가상 채널을 가진 라우터는 총 50개의 중재기가 필요하다.

$$N_{VA} = P_i \times P_o + P_o \times v \quad (4)$$

VC 라우터에서 SA도 VA와 마찬가지로 그림 4(b)와 같이 2단계의 중재 과정이 필요하다<sup>[5]</sup>. 첫 번째 단계에서는 각 입력 포트의 서로 다른 가상 채널에 존재하는 패킷의 스위치 접근 요청에 대한 중재를 수행한다. 두 번째 단계에서는 각 입력 포트에서 권한을 얻은 요청에 대하여 동일한 출력 포트에 통과하려는 패킷에 대한 중재를 수행한다. 2 단계 중재를 통해 스위치 접근 허락을 받은 패킷은 바로 크로스바와 출력 채널을 통하여 다음 라우터로 전달된다. VC 라우터에서 SA의 중재기 수는 수식 (5)과 같다. 5개의 입출력을 가진 라우터는 10개의 중재기가 필요하다.

$$N_{SA} = P_i + P_o \quad (5)$$

VC 라우터의 VA와 SA는 많은 수의 중재기가 필요할 뿐만 아니라 각각 2단계의 중재과정을 거쳐야 하기 때문에 VA와 SA를 단일 사이클에 동작시키는 것은 VC 라우터 전체의 동작 주파를 제한하는 요인이 된다. 따라서 라우터의 지연시간을 최적화하기 위해서는 VA

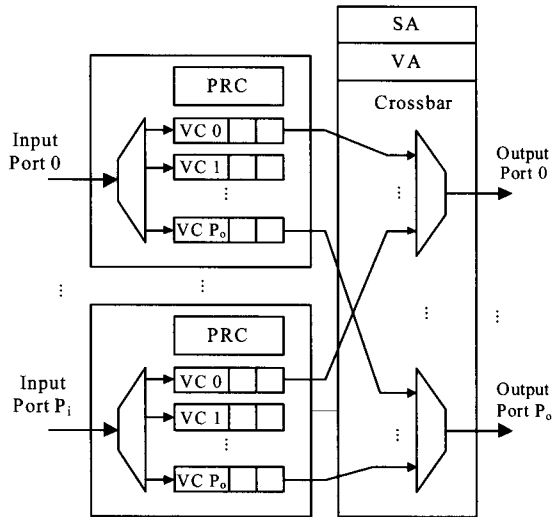


그림 5. SOLA NoC의 라우터 구조  
 Fig. 5. The router architecture in SOLA NoC.

와 SA의 복잡도를 최소화하는 것이 필요하다.

### 3. 라우터 구조

VC 라우터는 VA와 SA 복잡도는 VC 채널 개수에 따라 급격하게 증가한다. VC 채널에 따라서 VA와 SA 복잡도가 급격하게 증가하는 이유는 각 VC에 있는 패킷은 어떠한 출력 포트든지 선택하여 나갈 수 있고 출력 포트로 나간 패킷은 연결된 입력 포트에 존재하는 어떠한 VC 채널에도 저장 가능하다고 가정하기 때문이다. 이는 VA와 SA에 있는 중재기 개수를 증가시키고 각 중재기의 입력 및 출력 신호선을 증가시키기 때문에 VA와 SA의 복잡도는 급격하게 증가한다.

따라서 제안하는 라우터 구조는 각 버퍼로 들어오는 패킷이 출력되는 포트를 하나로 제한하였다. 이는 기존의 출력 버퍼 방식과 한 가지 차이점이 있다. 기존 출력 버퍼 방식은 출력 포트에 지원하는 VC의 개수만큼 버퍼를 두는 방식이지만, SOLA에서 사용하는 출력 버퍼 방식은 입력 포트에 출력 포트의 개수만큼의 버퍼를 두고 각 버퍼는 서로 다른 출력 포트와 연결되는 방식이다.

VC 라우터에서 Routing Computation (RC) 로직은 현재 입력 버퍼로 들어온 패킷이 어느 출력 포트로 나갈지를 결정한다. 제안한 라우터는 각 입력 버퍼에 들어온 패킷의 경로를 하나의 출력 포트 고정시켰기 때문에 기존의 VC 라우터와는 달리 라우팅 경로를 계산하는 로직은 다음 라우터에서 출력으로 나갈 방향을 미리 판단하여야 한다. 따라서 각 입력 버퍼는 출력 포트

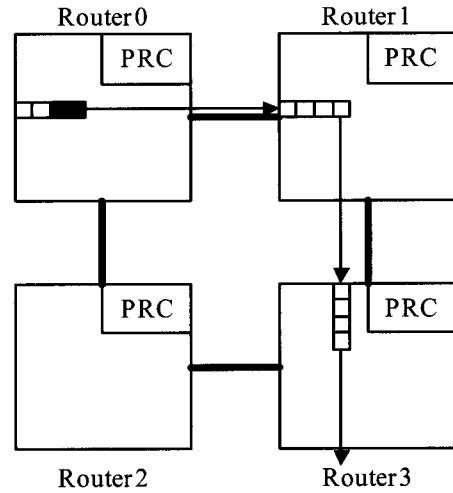


그림 6. PRC 동작을 설명하기 위한 NoC의 상태 예시  
 Fig. 6. The example for describing PRC mechanism.

로 나가기 전에 PRC를 이용하여 미리 다음 라우터의 출력 경로를 계산한다. 그림 5는 제안된 라우터의 구조를 나타낸다.

그림 6은 Router 0의 west 입력 포트에 있는 패킷이 Router 1과 Router 3을 통하여 Router 3의 north 출력 포트에 전송되는 것을 나타낸다. 패킷이 Router 0에서 east 출력 포트에 전달되어야 하기 때문에 Router 0의 라우팅 로직은 east를 연산 결과로 출력할 것이다. 하지만 Router 0의 PRC는 이미 현재 라우터에서 출력되는 방향은 이전 라우터에서 결정이 되어 미리 알고 있기 때문에 경로 상에 다음 라우터인 Router 1에서 출력 방향 (south)을 계산한다.

그림 7은 제안한 라우터의 VA 및 SA 구조를 보여주고 있다. VA와 SA 로직은 각 출력 포트마다 존재한다. VC 라우터와 달리 VA와 SA는 1단계로 구성된다. VA는 서로 다른 입력 포트에서 선택된 출력 포트를 통해 전달하려는 패킷 중에 같은 버퍼로 들어가려는 패킷 간의 중재를 수행한다. SA는 VA를 통해 선택된 패킷 중에서 출력 포트에 전달된 패킷을 최종적으로 선택한다.

패킷이 현재 통과하려는 라우터의 출력 포트 정보는 VA와 SA에서 필요한 정보이다. 출력 포트 정보가 통과하려는 라우터에서 계산하는 것인 VA와 SA 단계에 필요한 정보 계산하는 시간으로 인하여 지연시간이 증가한다. 하지만 PRC를 사용하면 이전 라우터에서 현재 통과하려는 라우터의 출력 포트를 계산하여 주기 때문에 VA와 SA를 바로 수행할 수 있다. 뿐만 아니라 경로 상의 다음 라우터에서 패킷이 출력되는 방향을 정하

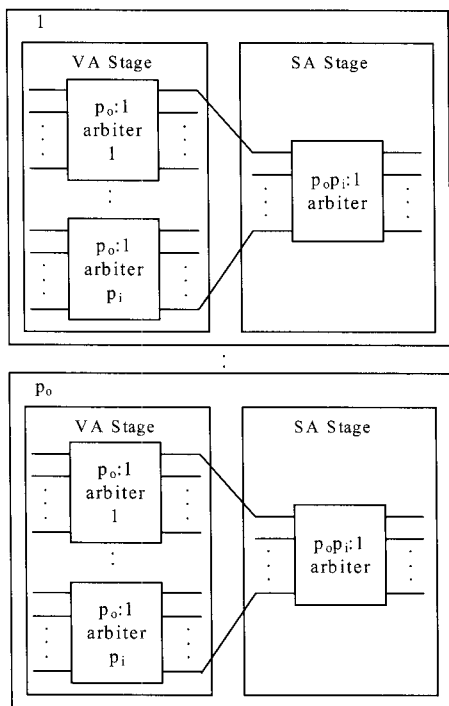


그림 7. SOLA NoC의 라우터의 VA 및 SA 구조  
Fig. 7. The VA and SA architectures of the router in SOLA NoC.

는 PRC는 현재 통과하려는 라우터에서는 VA와 SA 로 직과는 서로 의존관계가 없으므로 동시에 수행이 가능하다. 따라서 PRC를 사용하면 VC 라우터의 3단계 파이프라인을 2단계로 최적화 할 수 있다.

최초로 패킷이 라우터로 입력될 때는 입력된 라우터의 출력 포트는 네트워크 인터페이스에서 미리 계산한다. 마지막에 패킷이 최종 목적지 라우터에서 네트워크 인터페이스로 나오는 경우에 최종 목적지의 라우터에서는 이미 이전 라우터에서 최종 목적지의 라우터의 출력 포트를 결정하였기 때문에 PRC를 수행하지 않는다.

제안한 라우터는 VA 및 SA 복잡도를 줄이는 것이 가능하다. VA는 같은 출력 포트에 나가는 패킷 중에 같은 VC, 즉 다음 라우터에서 같은 방향으로 통과하는 패킷 간의 중재를 수행한다. VA는 각 출력 포트 당  $P_i$  개의 중재기가 필요하다. 각 중재기는 출력포트와 연결된 다음 라우터의 각 입력 버퍼에 전송되는 패킷 사이의 중재를 수행한다. SOLA 라우터의 총 중재기 수는 수식 (6)과 같다.

$$N_{VA} = P_i \times P_o \tag{6}$$

SA는 같은 출력 포트에 나가는 패킷 간의 중재를 수행한다. 입력 포트에 있는 버퍼는 각각 고정된 출력 포

트 방향으로만 패킷을 전달하기 때문에 출력 포트에는 하나의 중재기에 의해서 스위치 할당이 가능하다. 따라서 총 SA 중재기의 개수는  $P_o$ 이다.

제안한 라우터가 필요한 VA와 SA에서 필요한 총 중재기의 수는 수식 (7)과 같다. 5개의 입출력 포트와 5개의 가상 채널을 가진 VC 라우터 및 제안한 라우터의 총 중재기 수는 각각 60, 30개의 중재기가 필요하다. 따라서 제안한 라우터는 기존의 VC 라우터에 비해 중재기의 수를 반으로 줄일 수 있다.

$$N_{SOLA} = P_o \times P_i + P_o \tag{7}$$

뿐만 아니라 VC 라우터는 VA와 SA가 각각 2 단계로 구성되어 있기 때문에 입력으로 들어온 패킷을 전달하는데 총 4 단계의 중재 과정이 필요하다. 하지만 제안한 라우터는 VA와 SA 단계가 모두 단일 단계로 처리가능하기 때문에 패킷을 전달하는데 총 2단계의 중재 과정이 필요하다. 따라서 VA와 SA를 단일 사이클에 수행이 가능하다. 따라서 PRC는 VA와 SA와 독립적으로 동작 가능하기 때문에 입력 패킷에 대하여 출력 포트에 전달하는 과정에 필요한 지연시간은 중재과정에서 경쟁이 발생하지 않는 한 1 사이클에 패킷을 전달할 수 있다.

#### 4. 패킷 구조

SOLA NoC는 패킷 스위치 통신 구조이기 때문에 IP 코어 통신 프로토콜을 패킷으로 구성하여야 한다. SOLA에서 사용하는 패킷의 구조는 그림 8과 같다.

패킷은 크게 header와 payload로 구성된다. 3개의 word 정보로 구성된 header는 address와 control 정보를 포함하고 있어 전체 IP 코어 간의 통신 흐름을 결정한다.

첫 번째 word는 패킷을 생성한 IP 코어의 노드의 번

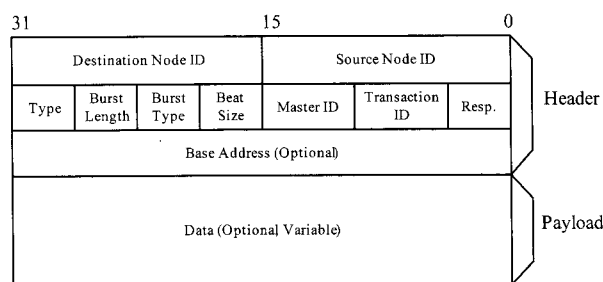


그림 8. SOLA NoC 사용하는 패킷 구조  
Fig. 8. Packet format for SOLA NoC.

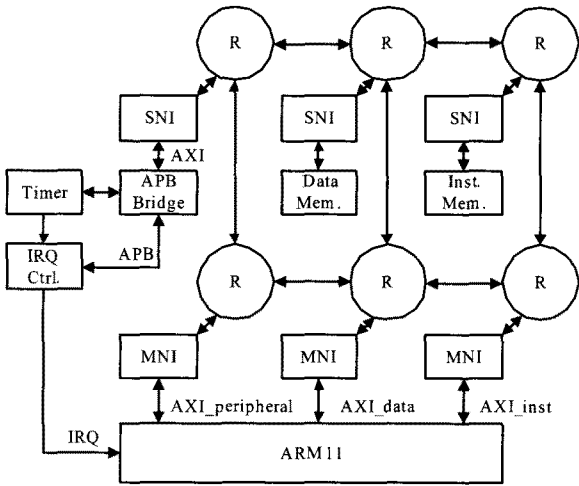


그림 9. SOLA NoC 기반의 가상 SoC 플랫폼  
Fig. 9. SoC virtual platform based on SOLA NoC.

호 (Source Node ID)와 목적지 노드의 번호 (Destination Node ID)로 구성된다. 목적지 노드 번호는 라우터에서 패킷 흐름을 결정하기 위하여 사용이 된다. XY 라우팅 알고리즘의 경우에는 목적지 노드 번호와 통과하려는 라우터의 번호를 비교하여 출력 포트를 결정한다. application-specific NoC를 위하여 routing table 기반의 라우팅 알고리즘을 사용하는 경우 routing table을 참고하기 위하여 목적지 노드 번호를 사용한다. 소스 노드 번호는 마스터 IP 코어에서 내보낸 요청 패킷에 대하여 슬레이브 IP 코어는 응답을 다시 돌려주어야 하기 때문에 요청 패킷의 소스 노드 번호는 응답 패킷의 목적지 노드 번호로 설정된다.

두 번째와 세 번째 word는 각각 IP 코어의 통신 프로토콜의 control과 address 정보로 구성된다. IP 코어의 통신 프로토콜의 control 정보는 일반적으로 read/write, burst length, burst type, burst beat size,

표 1. Flit이 속하는 패킷의 위치에 대한 2-bit 표기  
Table 1. The 2-bit representation for a position of flit in a packet.

2-bit 표기	첫 번째 flit	마지막 flit	나머지 flit
	01	11	10

master ID, transaction ID, response 정보를 포함하고 있다. 따라서 이와 같은 정보를 목적지 노드에 전달하고, 이를 재생해 목적지의 IP 코어에 전달할 필요가 있기 때문에 패킷 헤더에 control 정보를 포함한다. address 정보도 control 정보와 마찬가지로의 이유로 패킷 헤더에 포함되나, 응답 패킷의 경우에는 address 정보가 필요하지 않기 때문에 세 번째 word는 선택적으로 사용한다.

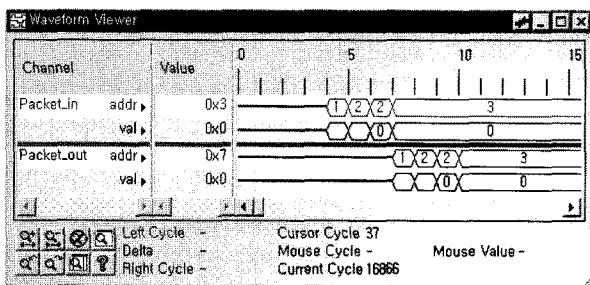
Payload에는 IP 코어의 통신 프로토콜에서 데이터 정보가 포함된다. 데이터는 마스터 IP 코어가 read 요청을 할 때나 슬레이브 IP 코어가 write 응답을 할 때에는 데이터가 필요하지 않기 때문에 선택적으로 payload를 사용한다.

패킷은 데이터의 크기가 가변적으로 변할 수 있기 때문에 패킷을 flit 단위로 전송하기 위해서는 패킷의 시작과 끝을 표시하는 것이 필요하다. 따라서 SOLA에서는 flit을 전송하는 채널에 2-bit의 side channel을 추가하여 flit이 패킷의 어느 부분인지를 표시한다. 표 1은 2-bit 정보에 따른 flit이 속하는 패킷의 위치에 대해서 나타내고 있다.

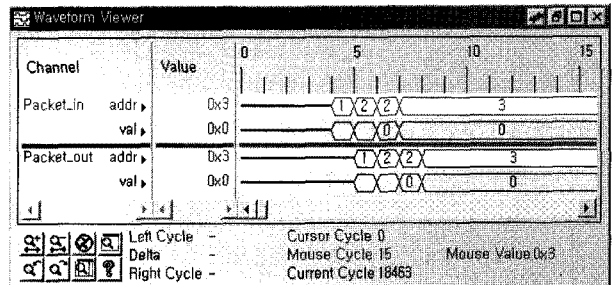
### V. 실험 결과

#### 1. SOLA NoC 기능 검증

SOLA NoC의 기능을 검증하기 위해 SoC Designer



(a)



(b)

그림 10. VC 라우터와 SOLA 라우터의 지연시간 비교, (a) VC 라우터의 waveform, (b) SOLA 라우터의 waveform  
Fig. 10. Latency comparison of VC router and SOLA router.

(a) the waveform of VC router, (b) the waveform of SOLA router

도구를 이용하여 그림 9와 같이 ARM11 프로세서, 메모리, Timer로 구성된 SOLA NoC 기반의 가상의 SoC 플랫폼을 구성하였다. SOLA NoC 기반의 가상 SoC 플랫폼의 검증은 Dhrystone<sup>[10]</sup> benchmark 프로그램에 의해 검증하였다.

SOLA를 구성하는 라우터의 지연시간을 확인하기 위하여 SoC Designer<sup>[8]</sup>에서 제공하는 Waveform Viewer 도구를 이용하였다. 그림 10은 VC 라우터와 SOLA 라우터의 동작에 대한 waveform을 보여주고 있다. VC 라우터는 패킷이 입력으로 들어온 뒤 3 사이클 이후에 패킷이 출력되는 것을 확인할 수 있다. 이는 3 단계 파이프라인으로 인하여 발생하는 지연시간 때문이다. 반면 SOLA 라우터는 PRC, VA, SA 모두 한 사이클에 동작 가능하기 때문에 패킷이 입력으로 들어온 뒤 1 사이클 이후에 패킷이 출력되는 것을 확인할 수 있다.

### 2. SOLA 라우터 성능 평가

SOLA 라우터 및 VC 라우터의 성능을 비교 평가하기 위하여 그림 11과 같이 4x4 mesh 구조의 SOLA NoC 시뮬레이션 플랫폼을 SoC Designer를 이용하여 구성하였다. 시뮬레이션을 위한 매개변수는 표 2와 같다. 각 노드에서 발생하는 트래픽은 미리 정해진 대역폭에 따라서 일정한 간격으로 고정 길이의 패킷으로 발생된다. 목적지 노드 선정은 uniform random 방식으로 목적지 노드를 선택하여 패킷을 보낸다. 시뮬레이션 데이터를 10만 사이클의 warm-up 시간 이후에 90만 사이클 동안 수집하여 성능 분석에 사용하였다.

그림 12는 VC 라우터와 SOLA 라우터를 이용한 NoC에서 발생한 평균 패킷 지연시간의 그래프이다. 그

표 2. 시뮬레이션 매개변수  
Fig. 2. Simulation parameters.

Simulation parameters	Specifications
Topology	4x4 2D mesh
Routing	Dimension-Order Routing (DOR)
Router latency (VC/SOLA)	3/1 cycles
Number of VCs per port	5
Buffer depth	8 flits
Packet size	4 flits/packet
Traffic pattern	Uniform random
Simulation cycle	1,000,000 cycles
Warmup cycle	100,000 cycles

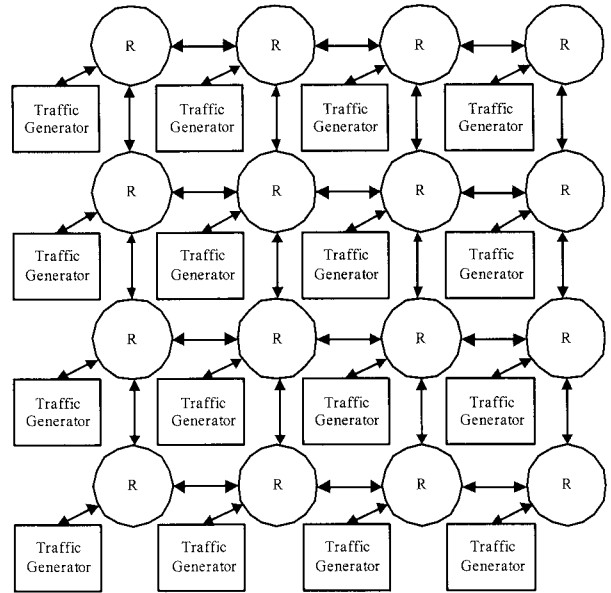


그림 11. SOLA 라우터 및 VC 라우터의 성능 평가를 위한 4x4 mesh NoC

Fig. 11. 4x4 mesh NoC architecture for evaluating performance of SOLA router and VC router.

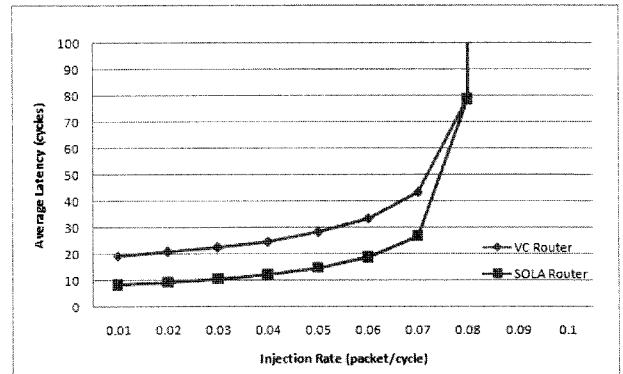


그림 12. VC 라우터와 SOLA 라우터의 평균 패킷 지연시간 비교 그래프

Fig. 12. The average packet latency graph of VC router and SOLA router.

래프의 x축은 사이클 당 입력된 패킷 수를 나타내고, y축은 평균 지연시간을 나타낸다. SOLA 라우터를 이용한 NoC에서 VC 라우터 기반의 NoC에 비해 평균 50%의 지연시간 감소를 갖고 왔다. 특히 낮은 트래픽 환경에서 SOLA 라우터 기반의 NoC는 높은 지연시간 감소를 얻을 수 있었다. 반면 높은 트래픽이 발생한 혼잡한 환경에서는 VC 라우터와 SOLA 라우터는 거의 유사한 지연시간을 보이고 있다. 이는 IV장의 수식 (2)에 의해 해석할 수 있다. 높은 트래픽 환경에서는 패킷 간의 채널 접근 경쟁으로 발생하는 지연시간  $T_{router\_contention}$ 가 라우터 파이프라인에 의해 발생하는 지연시간



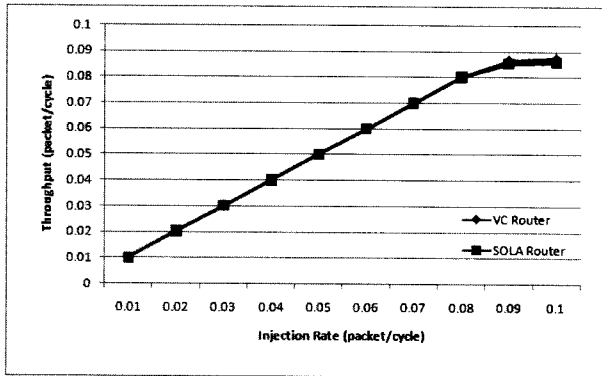


그림 13. VC 라우터와 SOLA 라우터의 처리량 비교 그래프

Fig. 13. The throughput graph of VC router and SOLA router.

$T_{router\_pipe}$  보다 상대적으로 커지기 때문이다. 반면 SOLA 라우터는  $T_{router\_pipe}$  을 감소시키는 구조이기 때문에 높은 트래픽 환경에서는 상대적으로 커진  $T_{router\_contention}$  으로 인하여 두 NoC는 거의 같은 지연시간을 보이는 것이다.

그림 13은 VC 라우터 기반의 NoC와 SOLA 라우터 기반의 NoC의 throughput을 비교한 그래프이다. 그래프의 x축은 사이클 당 입력된 패킷 수를 나타내고, y축은 사이클 당 출력된 패킷 수로서 패킷 처리량인 throughput을 나타낸다. SOLA 라우터 기반의 NoC는 throughput은 VC 라우터 기반의 NoC에 비해 약 1-2% 정도 낮은 성능을 보였다. 하지만 이는 지연시간에서 얻을 수 있는 이득을 감안한다면 거의 무시할만한 수준이다.

## VI. 결 론

SoC에 집적되는 IP 수의 증가와 높은 통신 대역폭을 요구하는 애플리케이션의 등장함에 따라 SoC 통신 구조는 패킷 스위치 기반의 NoC로 진화하고 있다. 하지만 기존 VC 라우터의 높은 컨트롤 복잡도로 인하여 입력으로 들어온 패킷을 출력 포트에 전달하는데 3 단계의 파이프라인을 거쳐야하기 때문에 최소 3 사이클의 지연시간이 발생한다.

따라서 본 논문에서 제시하는 SOLA는 라우터를 최적화하는 구조를 제시함으로써 NoC에서 발생하는 지연시간을 최소화하였다. 라우터는 각 VC마다 출력 가능한 포트를 하나로 제한하였다. 그리고 이러한 구조는 라우팅 구조를 변경하여야 하기 때문에 미리 다음 라우

터에서의 경로를 계산하는 PRC 로직을 설계하였다. 이러한 구조는 기존 파이프라인 단계를 단일 단계로 최적화하여 입력으로 들어온 패킷이 출력 포트에 나가는데 채널 경합이 없는 경우 한 사이클이 발생하도록 지연시간을 최소화하였다.

SOLA NoC 구조의 기능 검증 및 성능 평가를 위하여 상위 레벨 모델링 방법인 ESL 모델링 방법을 이용하여 SOLA 라우터와 네트워크 인터페이스를 모델링하고 SoC Designer 도구를 이용하여 시뮬레이션 환경을 구성하였다. 시뮬레이션 환경을 통하여 실제 애플리케이션을 동작시킴으로서 SOLA 라우터와 네트워크 인터페이스를 이를 검증하였다. 또한 성능 평가 결과 VC 라우터 기반의 NoC에 비하여 SOLA NoC는 평균적으로 50%의 지연시간 감소를 나타냈다.

## 참 고 문 헌

- [1] Andrei Radulescu, et al., "An Efficient On-Chip Network Interface Offering Guaranteed Services, Shared-Memory Abstraction, and Flexible Network Programming," IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, No. 1, pp. 4-17. Jan. 2005.
- [2] Douglas Densmore, Roberto Passerone, Alberto Sangiovanni-Vincentelli, "A Platform-Based Taxonomy for ESL Design," IEEE Design and Test of Computers, Vol. 23, No. 5, pp. 359-374, Sept. 2006.
- [3] Eui Bong Jung, et al., "SONA: An On-Chip Network for Scalable Interconnection of AMBA-Based IPs," International Conference on Computational Science, Vol. 4. pp. 244-251, 2006.
- [4] L. Benini, et al., "Networks on Chips: a New SoC Paradigm," IEEE Computer, Vol. 35, No. 1, pp.70-78 Jan. 2002.
- [5] Li-Shiuan Peh, et al., "A Delay Model and Speculative Architecture for Pipelined Routers," in Proceedings of International Symposium on High-Performance Computer Architecture, pp. 255-266, Nuevo Leone, Mexico, Jan. 2001.
- [6] Seong Min Jo, Jin Seok Ha, and Yong Ho Song, "Design and Implementation of Network Interface with Layered Pipeline Architecture," in Proceedings of the 7th International Conference on Applications and Principles of Information Science, pp. 318-321, Auckland, New Zealand, Jan. 2008.

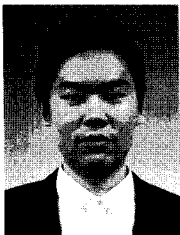
[7] M. T. Rose, The Open Book: A Practical Perspective on OSI, Prentice Hall, Englewood Cliffs, NJ, 1990.

[8] Carbon SoC Designer, <http://carbondesignsystems.com/>

[9] W. J. Dally, Principles and Practices of Interconnection Networks, Morgan Kaufmann, 2004.

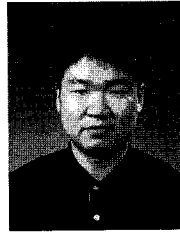
[10] Reinhold P. Weicker, "Dhrystone Benchmark: Rationale for Version 2 and Measurement Rules," SIGPLAN Notices Vol. 23, No. 8, pp. 49-62, Aug. 1998. 8.

저 자 소 개



조 성 민(학생회원)  
 2006년 한양대학교 정보통신학부  
 학사 졸업.  
 2008년 한양대학교  
 전자컴퓨터통신공학과  
 석사 졸업.

<주관심분야: NoC, SoC, 멀티코어시스템>



조 한 욱(학생회원)  
 1992년 서울대학교 컴퓨터공학과  
 학사 졸업.  
 1994년 서울대학교 컴퓨터공학과  
 석사 졸업.  
 2002년 University of Southern  
 California, Electrical  
 Engineering, 석사 졸업.

<주관심분야: 임베디드시스템, 멀티미디어, WLAN>



하 진 석(학생회원)  
 2001년 인하대학교 재료공학부  
 학사 졸업.  
 2003년 인하대학교 정보통신학부  
 석사 졸업.  
 <주관심분야: 임베디드시스템,  
 SoC>



송 용 호(정회원)  
 1989년 서울대학교 컴퓨터공학과  
 학사 졸업.  
 1991년 서울대학교 컴퓨터공학과  
 석사 졸업.  
 2002년 University of Southern  
 California, Electrical  
 Engineering, 박사 졸업.

<주관심분야: 임베디드시스템, SoC, NoC>