

미세 단위 클럭 조정을 통한 저전력 NoC 라우터 구조

이한울<sup>1</sup>, 이병훈<sup>1</sup>, 정의영<sup>1</sup>  
<sup>1</sup>연세대학교 전기전자공학부

A Low-Power NoC Router Architecture with Fine-Grained Clock Control

Hanul Lee<sup>1</sup>, Byunghoon Lee<sup>1</sup>, Eui-Young Chung<sup>1</sup>

<sup>1</sup>School of Electrical and Electronic Engineering Yonsei University

**Abstract** - As NoCs are largely adopted in handheld devices, researches for low-power NoC become necessary. We expect that power consumption in NoCs can be minimized by assigning appropriate clock frequency to each link based on the fact that different amount of throughput is required to each channel. In this paper, we propose a novel NoC router architecture which is able to reduce power consumption not only on links but also in router internals. We compared our proposed router with the conventional synchronous ones in a conservative way. The results show that up to 20% power saving can be achieved with our router.

1. 서 론

공정 기술의 발달은 하나의 칩 안에 더 많은 트랜지스터를 구현할 수 있게 함으로써, 다수의 IP(Intellectual Property)들을 단일 칩에 구현하는 SoC(System on Chip)의 형태로 발전하였다. 단일 칩 안에 구현되는 PE(Processing Element) 수의 증가는 더 높은 bandwidth를 제공할 수 있는 on-chip interconnection을 요구하게 되었다. 이는 전통적인 bus 구조의 한계를 극복할 수 있는 NoC(Networks on Chips)[1]의 발전으로 이어졌으며, NoC는 이제 휴대용 단말기에까지 적용되는 시대에 이르렀다.

지난 십여 년간 NoC에 관한 많은 연구들이 진행되었으나, 그들은 대부분 성능 개선에 그 초점이 맞춰져 있었다. 하지만 배터리를 기반으로 하는 휴대용 기기들에서도 높은 성능의 SoC가 요구됨에 따라 저전력 NoC에 대한 연구는 필수적인 것으로 여겨지고 있다.

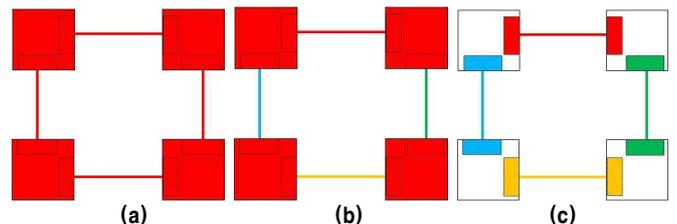
저전력 NoC에 대한 연구는 전력 최적화를 위한 대상의 범위로 구분할 수 있다. NoC는 일반적으로 NI(Network Interface), 라우터, 링크에 의해서 구성된다고 할 수 있는데, NI는 IP의 특성에 의존적이기 때문에 저전력 NoC에 대한 연구 범위로 적당하지 않다. 라우터 내부에서 소모 전력을 최소화할 수 있는 방법으로는 crossbar 부분 활성화, 입력 버퍼 구조, 저전력 arbiter에 대한 연구들[2-3]이 대표적이다. 하지만 이들은 물리적인 구현 수준에 집중되어 있어 네트워크 traffic 등을 고려하는 등의 시스템 수준에서의 고찰에 대한 한계가 있다. 링크 전력을 줄이기 위한 연구로는 링크의 사용 빈도를 바탕으로 클럭 recovery 기술 등을 활용하여 링크 단위에서 클럭을 조정하는 연구[4]가 대표적이다. [4]는 시스템 관점에서 네트워크 특성을 활용하여 전력을 최소화하려고 하였으나, 라우터 내부의 전력 소모를 줄이지 못 하는 단점이 있다.

대다수의 SoC들은 그 사용 목적에 특수화되어 설계되는 Application-specific MPSoC(Multi-Processor SoC)의 형태를 갖는데, 이러한 SoC를 구성하는 IP들은 서로 다른 특성을 갖기 때문에 NoC를 구성하는 각 링크별로 서로 다른 데이터 처리량(throughput)이 요구된다. 본 논문에서는 NoC의 각 링크별로 요구되는 throughput에 최적화된 클럭 주파수 할당을 통해서 전력 소모를 최소화할 수 있는 NoC를 위한 라우터 구조를 제안한다.

2. 본 론

2.1 Motivation

각 PE가 독립적인 클럭 및 전압 영역(voltage-frequency island)에서 동작하는 특성을 갖는 MPSoC에서의 NoC는 그 뼈대가 되는 라우터에 대한 클럭 할당 방식에 따라 크게 두 가지로 분류할 수 있다. 첫 째는 각 라우터가 귀속된 클럭 및 전압 영역의 클럭을 사용하는 것이다. 이 방식은 클럭 네트워크 합성(clock tree synthesis) 및 소모 전력 측면에서 유리한 점이 있다. 하지만 이는 on-chip 네트워크와 IP들을 분리하는 NoC의 개념에 맞지 않으며, 특정 네트워크 노드(node)에서 성능이 크게 저하되는 단점이 있다. 두 번째 방식은 on-chip 네트워크를 위한 별도의 클럭을 할당하여 네트워크를 구성하는 모든 라우터와 링크가 같은 속도로 동작하게 하는 것이다. 이 경우 on-chip 네트워크와 IP들을 분리하여 설계할 수 있으며, IP들의 클럭 속도와 상관없이 클럭 속도를 결정할 수 있기 때문에 성능 저하를 막을 수 있다. MPSoC의 NoC에서는 일반적으로 두 번째 방법인 별도의 네트워크 클럭을 할당하는 방식이 일반적이다.



<그림 1> NoC 라우터 전력 소모 비율[8]

<그림1>은 on-chip 네트워크가 별도의 클럭에 의해서 동작할 때, 전력을 최적화할 수 있는 3종류의 클럭 할당 방식을 2\*2 mesh topology의 예시를 통해서 보여준다. 4개의 큰 사각형, 내부의 작은 사각형, 직선은 순서대로 각각 네트워크 노드 또는 라우터, 라우터 포트, 링크를 의미한다. 각 구성요소는 그것을 동작하는 클럭 속도에 따라 서로 다른 색깔들로 표현되어 있다. <그림1-a>는 앞서 설명한 모든 네트워크 구성 요소들이 같은 클럭 속도로 동작하는 방식을 보여준다. 이 경우 클럭 속도는 모든 네트워크 링크 중에서 가장 큰 처리량을 요구하는 링크에 맞춰지는 것이 보편적이기 때문에 불필요하게 높은 클럭 속도를 할당받는 링크가 생기게 된다. 반대로 소모 전력을 줄이기 위해서 낮은 클럭 속도로 동작하게 될 경우 성능 저하가 일어날 수 있다. 다시 말해서, 여러 종류의 IP들이 통신하는 MPSoC에서는 링크별로 서로 다른 처리량이 요구되는데 일괄적인 클럭 속도를 할당하게 되면 전력 및 성능 면에서 효과적이지 못 하다. <그림1-b>는 앞서 설명한 링크별 throughput 요구량에 대한 비등가성에 따른 비효율성을 극복하기 위해서 링크 단위에서 클럭을 조정하여 소모 전력을 줄이기 위한 연구들 [4-5]을 보여준다. 이들은 클럭 조정 방식에 따라서 동적 또는 정적이라는 점의 차이가 있으나, 라우터 내부의 동작 클럭을 고정된 채 링크 클럭만을 조정하였다는 공통점이 있다. [6]에 따르면 NoC의 전력 소모는 클럭 네트워크(33%), FIFO 버퍼(22%), 링크(17%) 등의 순으로 이루어지는 것을 알 수 있는데, <그림1-b>와 같은 클럭 할당 방식을 통해서 링크보다 더 큰 부분을 차지하는 라우터 내부의 FIFO 버퍼 등에 의한 전력 소모를 줄일 수 없다. 본 논문에서는 <그림1-c>와 같은 클럭 할당 방식을 적용할 수 있는 라우터 구조를 제안한다. 각 링크별 클럭 속도는 설계 단계에서 얻어진 throughput 요구량에 의해서 결정된다. 또한, 라우터 전체 단위의 클럭이 아닌 포트별로 분산된 클럭을 할당하는 구조를 통해서 각 포트가 연결된 링크에서 필요로 하는 속도에 비례하여 동작하도록 하여 라우터 내부의 전력 소모를 줄인다.

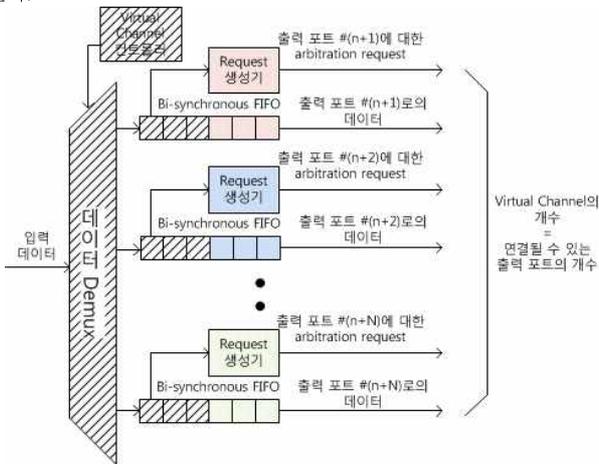
2.2 Router Architecture

본 논문은 라우터에 연결된 각각의 링크가 서로 다른 클럭 주파수에 의해서 동작하는 것을 가정한다. 또한 링크에서 요구되는 처리량에 맞춰서 라우터 내부의 버퍼 등이 동작하도록 하여 라우터 내부의 전력 소모를 최소화하고자 하였다. 이를 위해서 라우터를 구성하는 sub-module들을 포트 단위로 분산하여 설계함으로써, 라우터 내부 동작 전체를 관장하는 단일 클럭을 제거하였다. 이러한 방식의 설계를 통해서 라우터를 다양한 크기로 확장시켜 regular topology뿐만 아니라 irregular topology에도 적용시킬 수 있다. 본 논문에서 제안하는 라우터는 worm-hole flow control을 사용하며, deterministic 라우팅 방식을 이용한다. 다음에서는 각각의 입/출력 포트의 구조를 설명한다.

2.2.1 입력 포트

<그림2>는 제안하는 라우터의 입력 포트 구조를 보여주며, 각각의 클럭 영역(clock domain)은 다른 색으로 표시되어 있다. 본 논문에서는 라우터의 모든 포트가 서로 다른 클럭으로 동작하는 것을 가정한다. 이 경우, 각 입력 포트는 하나의 쓰기 클럭과 여러 개의 읽기 클럭으로 이루어지는데 서로 다른 클럭 영역 간의 통신을 위해서는 synchronizer가 필요하다. 입력 포트를 단일 채널과 둘 이상의 읽기 클럭에 의해서 동작하는 multi-synchronous FIFO를 통해서 구성하는 경우, 다수의 읽기 클럭을 선택하기 위한 클럭 스위치가 필요하다. 클럭 스위치는 일반적으로 선택하는 클럭이 바뀌게 되는 경우 2 cycle의 synchronization delay를 요구한다[7].

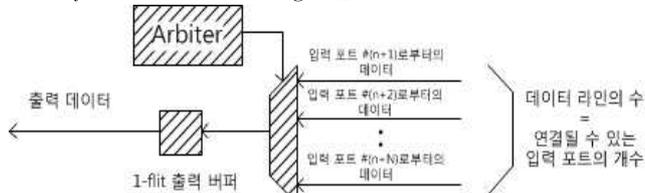
본 논문에서는 연속적으로 출력 포트가 바뀌는 데이터에 대비하기 위해서 각 입력 포트에서 연결될 수 있는 출력 포트 개수만큼의 virtual channel을 구성하고, 이들을 bi-synchronous FIFO로 연결하였다. 또한, 각각의 virtual channel은 출력 포트의 arbitration을 요청하기 위한 generator를 가지며, 각각의 request generator는 연결된 입기 또는 출력 포트의 클럭에 의해서 동작한다.



<그림 2> 라우터 입력 포트 내부 구조

2.2.1 출력 포트

<그림3>은 제안하는 라우터에서 출력 포트의 구조를 보여준다. 모든 출력 포트는 하나의 arbiter와 1-flit을 저장할 수 있는 출력 버퍼를 갖는다. 소모 전력을 최소화하기 위해서 round-robin 방식의 mux-tree arbiter[3]가 사용되었으며, 각 출력 포트에 출력 버퍼를 둬으로써 연결된 링크에서의 wire delay와 라우터 내부의 timing을 분리하였다.



<그림 3> 라우터 출력 포트 내부 구조

3. 실험

3.1 실험 환경

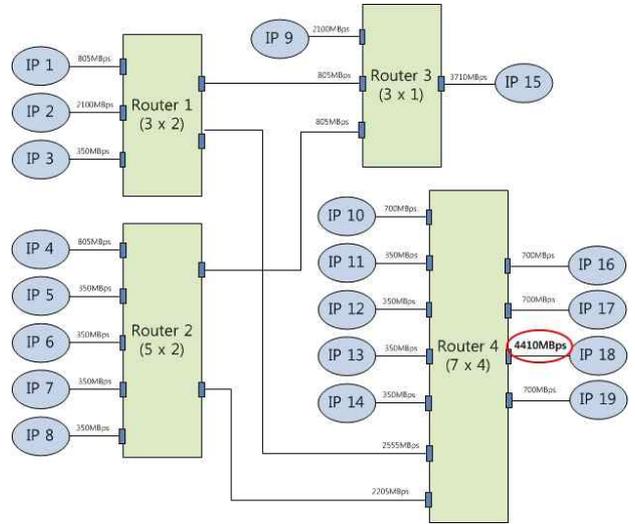
MPSoC 환경에서 제안하는 구조의 라우터에 대한 평가는 기존의 동기식 라우터와의 비교를 통해서 이루어졌다. 실험을 위해서 라우터들은 Register Transfer 수준으로 구현되었으며, 실험은 Synopsys Design Compiler를 사용하여 90nm 공정으로 합성된 후 전력 예측 값을 비교하는 방식으로 진행되었다. 또한, 실제의 휴대용 단말에서의 MPSoC에 가까운 실험을 위해서 [8]을 통해서 얻어진 3중류의 irregular topology를 기준으로 실험을 진행하였다. <그림4>는 실험에서 사용된 환경 중 하나인 “mobile application processor”의 topology 및 각각의 채널에서 요구되는 throughput을 보여준다. 또한, 성능에 의한 영향을 최소화하기 위해서 모든 라우터는 채널별로 동일하게 4 flit 크기의 입력 버퍼를 갖도록 설정하였다.

3.2 실험 결과

<표1>는 SoC 단위의 전력 소모 결과를 보여주며, 제안하는 구조의 라우터를 통해서 최대 20%까지 전력을 줄일 수 있는 것을 알 수 있다. <그림 5>는 “mobile application” 환경에서 각각의 라우터에서 소모되는 전력을 보여준다. 우리는 여기에서 “Router 4”에서 기존의 동기식 라우터에 비해서 더 많은 전력이 소모되는 것을 알 수 있다. 이는 본 논문에서 제안하는 라우터 구조의 특성상 라우터의 크기가 커지면 하나의 입력 포트에 구성되는 virtual channel의 숫자가 많아지면서 입력 버퍼의 크기가 커지기 때문이다. 본 논문에서의 실험은 각각의 채널별로 같은 크기의 버퍼 depth를 갖게끔 설정하였기 때문에 특정 포트에서 연결될 수 있는 출력 포트의 개수가 늘어난다면 라우터에서의 버퍼 크기가 비대칭적으로 커질 수 있다. 본 논문에서는 성능에 영향을 주지 않고자 채널별로 버퍼의 크기를 고정하였으나, 각 포트별 버퍼의 크기를 일정하게 하여 실험하거나 topology 합성 단계에서 이러한 특성을 고려한다면 더 좋은 결과를 얻을 수 있을 것으로 생각된다.

4. 결론

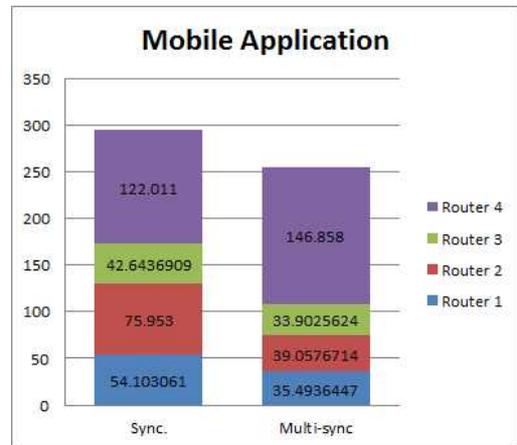
본 논문에서는 채널별로 동작 클럭 속도를 다르게 할 수 있는 NoC 라우터 구조를 제안하였다. 제안하는 구조의 NoC 라우터의 활용은 채널 단위로 다른 양의 throughput이 요구되는 MPSoC의 저전력화를 이끌어낼 수 있는 뼈대가 될 수 있을 것으로 생각된다. 또한, 제안한 라우터의 특성을 기존의 topology 합성 연구[8]에 활용한다면 더욱 효과적인 NoC를 얻어낼 수 있을 것으로 기대된다.



<그림 4> “Mobile Application Processor” 환경

	기존 동기식 NoC	제안하는 구조	감소 비율
mobile application processor	294.71mW	255.31mW	13%
mobile multimedia player	225.71mW	179.99mW	20%
mpeg4 decode	155mW	134.16mW	14%

<표 1> SoC 단위에서의 소모 전력 비교



<그림 5> “Mobile Application Processor” 에서의 라우터별 소모 전력 비교

[참고 문헌]

[1] L. Benini et al., “Networks on Chips: A New SoC Paradigm”, IEEE Computer, vol. 35, no. 1, pp. 70-78, 2002  
 [2] H. Wang et al., “Power-driven Design of Router Microarchitectures in On-chip Networks”, in Proc. IEEE/ACM International Symposium on Microarchitecture, pp. 105-116, 2003  
 [3] K. Lee et al., “Low-Power Network-on-Chip for High-Performance SoC Design”, IEEE Transactions on VLSI Systems, vol. 14, no. 2, pp. 148-160, 2006  
 [4] L. Shang et al., “Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks”, in Proc. International Symposium on High-Performance Computer Architecture, pp. 91-102, 2003  
 [5] D. Shin et al., “Power-Aware Communication Optimization for Network-on-Chips with Voltage Scalable Links”, in Proc. International Conference of Hardware/Software Codesign and System Synthesis, pp. 170-175, 2004  
 [6] Andrew B. Kahng et al., “ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration”, in Proc. Conference on Design, Automation and Test in Europe, pp. 423-428, 2009  
 [7] R. Ginosar, “Fourteen Ways to Fool Your Synchronizer”, in Proc. International Symposium on Asynchronous Circuits and Systems, pp. 89-96, 2003  
 [8] M. Jun et al., “Design of On-Chip Crossbar Network Topology Using Chained Edge Partitioning”, The Computer Journal, vol. 53, no. 7, pp. 904-917, 2010