

동적 라우팅을 사용하는 클러스터 기반 MPSoC 구조

김장억, 김재환, 안병규, 신봉식, 정정화
한양대학교 정보통신학과

Dynamic On-Chip Network based on Clustering for MPSoC

Jang-Eok Kim, Jae-Hwan Kim, Byung-Gyu Ahn, Bong-Sik Sin, Jong-Wha Chong
The College of Information & Communications
Hanyang University

E-mail : kim_100000000@hotmail.com, corekjh@ihanyang.ac.kr, mon809@chol.com,
seagiga@naver.com, jchong@hanyang.ac.kr

Abstract

Multiprocessor system is efficient and high performance architecture to overcome a limitation of single core SoC. In this paper, we propose a multiprocessor SoC (MPSoC) architecture which provides the low complexity and the high performance. The dynamic routing scheme has a serious problem in which the complexity of routing increases exponentially. We solve this problem by making a cluster with several PEs (Processing Element). In inter-cluster network, we use deterministic routing scheme and in intra-cluster network, we use dynamic routing scheme. In order to control the hierarchical network, we propose efficient router architecture by using smart crossbar switch. We modeled 2-D mesh topology and used simulator based on C/C++. The results of this routing scheme show that our approach has less complexity and improved throughput as compared with the pure deterministic routing architecture and the pure dynamic routing architecture.

신 오버헤드를 감당하지 못하기 때문에, 프로세서의 개수가 늘어나도 전체 성능이 선형적으로 증가하지 않는다.

최근 on-chip bus 구조를 대체하는 NoC (Network-on-chip)가 제안되었고, 네트워크를 잘 관리하는 것이 전체 시스템의 성능에 지대한 영향을 미치게 되었다. 본 논문에서 우리는 다수의 프로세서를 위한 효율적이며 뛰어난 확장성을 갖는 동적 라우팅 구조를 제안한다. 기존의 동적 라우팅 구조는 정적 라우팅에 비하여 복잡도가 매우 컸다. 이를 해결하기 위해 우리는 2-D mesh 구조에서 다수의 프로세서들을 묶어서 클러스터를 구성하였고, 이 클러스터와 클러스터 사이에서 효율적인 라우팅이 가능한 스위치 구조를 설계하였다.

I. 서론

최근, 반도체 기술의 눈부신 발전에 의하여 하나의 실리콘 다이에 수십억 개의 트랜지스터가 집적되고 있으며, SoC 시스템은 점점 복잡해지고 있다. 특히 응용 시스템의 복잡도가 크게 증가하여, 개발자들은 SoC 에 더 높은 성능을 갖는 프로세서를 요구하고 있다. 그러나 [1,2]에서 보여지듯이 프로세서의 클럭 속도를 높이는 것은 물리적인 한계가 있다. 따라서 전체 시스템의 처리 능력(throughput)을 높이기 위하여 하나의 SoC 안에 여러 개의 프로세서를 갖는 구조가 제안되고 있다. 그러나 기존의 on-chip bus 구조로는 프로세서간의 통

II. 본론

동적 라우팅은 결함 허용, 처리 속도 측면에서 정적 라우팅 보다 매우 효율적인 방법이다. 그러나 노드의 개수가 증가함에 따라 그 복잡도는 기하급수적으로 증가하게 되는 문제점을 갖고 있다. 우리는 2-D mesh 구조에서 클러스터링을 통하여 동적 라우팅에 따른 복잡도의 제한을 걸고 관리하였다. 그림 2 에 제시된 것처럼 클러스터와 클러스터 사이는 크로스바 스위치를 통하여 연결되어 있다.

기본적으로 클러스터 간의 통신은 정적 라우팅을 사용한다. 통신은 크로스바 스위치를 통하여 이루어진다. Packet 이 들어오면 크로스바 스위치가 갖고 있는 전

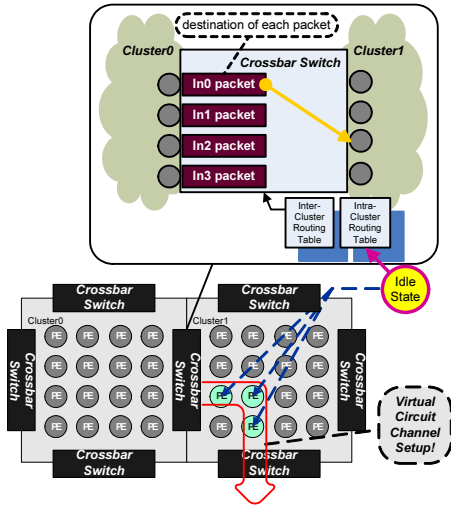


그림 1. 클러스터 간의 스위치 구조

역 라우팅 테이블을 이용하여 어느 방향으로 보낼지를 결정하게 된다. 들어온 패킷 헤더의 목적지 정보를 통하여 다음에 이동할 클러스터(east, west, south, north)를 정한다[3]. 클러스터간의 PE 들을 스위치가 연결하여 줄 때, 효율성을 높이기 위하여 그림 1 과 같이 동적으로 연결하여 준다. 클러스터내부의 통신시에는 동적 라우팅을 사용한다. 클러스터내부의 PE 들의 상태 정보를 스위치들이 이용하여 내부 클러스터를 위한 라우팅 테이블을 구성하게 된다. 스위치를 통하여 들어온 패킷은 내부 클러스터 라우팅 테이블을 이용하여 다음 스위치까지 갈 경로를 찾게 된다.

각 클러스터에 속한 스위치들은 PE 들로부터 주기적으로 상태정보를 받아 내부 라우팅 테이블을 업데이트한다. 각 PE 들은 n-bit 의 상태정보를 각 스위치에게 보내주게 된다. 1-bit signal 정보만으로는 라우팅의 효율성에 한계가 있을 수 밖에 없기 때문에 전체적인 네트워크의 상태를 파악하는 데에 한계가 있다. 다수 bit 의 정보를 갖고 있는 상태 패킷을 이용하여야만 전체 네트워크의 효율성을 기대할 수 있다.

그림 1 에 제시된 것처럼 라우팅 정보를 그 클러스터에 속한 4 개의 스위치들만이 갖고 있다. 기존의 각 노드들이 라우팅 테이블을 만들어 갖고 있는 방법은 각 노드들 간의 통신 복잡도가 상당히 높고 각 노드들이 각자의 라우팅 테이블을 갖기 때문에 계산 복잡도가 높아진다. 또한 [4]에서 보여지듯이 칩 내의 저장 자원을 소비하게 된다. 그러나 제안하는 방법은 해당 클러스터에 속한 스위치만이 라우팅 테이블을 갖고 있어서 통신 및 계산 복잡도를 최소화 할 수 있다.

III. 실험결과

그림 2 에 제시된 것처럼 제안하는 방식이 기존의 동적 라우팅 방법보다는 현저히 낮은 복잡도를 나타내었지만 성능은 비슷함을 나타내었고, 정적 라우팅 방법보다는 높은 복잡도를 가졌지만 높은 성능을 보였다.

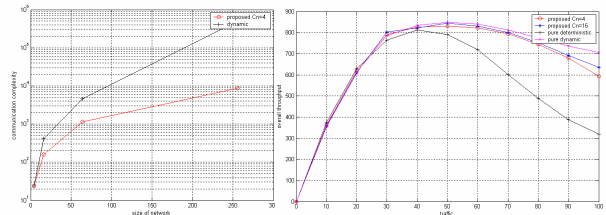


그림 2. 정적·동적 라우팅 방법과의 비교
(가) 라우팅 복잡도 (나) 전체 성능

IV. 결론

본 논문에서는 낮은 복잡도와 높은 성능을 갖는 새로운 멀티프로세서 SoC 구조를 제안하였다. 그리고 제안하는 구조는 뛰어난 결함 허용과 확장성을 갖는다. 실험결과 통신상의 비교적 낮은 복잡성과 높은 성능을 보였다.

감사의 글

본 논문은 정보통신부 정보통신연구개발사업의 지원에 의한 것임.

참고문헌

- [1] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," IEEE Computer, vol. 35, no. 1, pp. 70-78, Jan. 2002.
- [2] Bertozzi D., Benini L., "Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip," IEEE Circuits and Systems Magazine, pp.18-31, 2004
- [3] E. Beigne, F. Clermidy, P. Vivet, A. Clouard, M. Renaudin, "An asynchronous NOC architecture providing low latency service and its multi-level design framework," Proceedings of 5th Asynchronous Circuits and Systems(ASYNC'05), pp. 54-63, 2005.
- [4] Andrei Radulescu and Kees Goossens, "Communication Services for Networks on Chip," SAMOS, II, pp. 275-299, 2002.