논문 04-29-6B-5 한국통신학회논문지 '04-6 Vol.29 No.6B

위상 결합을 기반으로한 연결 망 설계 및 시뮬레이션

정회원 장 창 수*, 최 창 훈**

Design and Simulation of Interconnection Network Based on Topological Combination

Chang-soo, Jang, Chang-hoon, Choi *Regular Members*

요 약

본 논문에서는 정적 네트워크 위상과 동적 위상을 결합한 새로운 부류의 MIN(Multistage Interconnection Network)인 Combine MIN을 제안한다. Combine MIN은 단일 경로 성질을 갖는 MIN보다도 적은 하드웨어 비용을 가지면서도 다중 경로를 제공한다. 또한 Combine MIN은 빈번한 통신을 갖는 프로세서-메모리에 짧은 경로의 지름길 경로 및 다중 경로를 제공함으로써 지역화된 통신에 적합하게 구성할 수 있게 설계되었다. 성능 평가를 위한 시뮬레이션 결과에 따르면 Combine MIN은 높은 지역화된 통신에서 같은 네트워크 크기를 갖는 기존의 MIN보다 우수한 성능을 보였다. 따라서 Combine MIN은 공유 메모리 다중 프로세서 시스템에서 지역화된 통신 구조를 갖는 병렬 응용 분야에서 효율적으로 활용될 수 있을 것이다.

ABSTRACT

In this paper, we propose a new class of MIN(Multistage Interconnection Network) called Combine MIN which combines static network topology and dynamic network topology. Combine MIN provides multiple paths at a hardware cost lower than that of MIN with unique path property. Combine MIN can be constructed suitable for localized communication by providing the shortcut path and multiple paths inside the processor-memory cluster which has frequent data communications. According to the results of analysis and simulation for performance evaluation, Combine MIN shows higher performance than MINs of the same network size in the highly localized communication. Therefore, Combine MIN can be used as an attractive interconnection network for parallel applications with a localized communication pattern in shared-memory multiprocessor systems.

1. 서 론

일반적으로 공유 메모리 다중 프로세서 시스템 환경 하에서 많은 사용자들이 사용하는 대다수의 응용 프로그램들에서는 적은 수의 프로세서-메모리 그룹(group)내에서의 통신되는 빈도수는 전체 통신 양 중의 많은 부분을 차지하기 때문에, 이들 그룹에 대한 짧은 경로의 제공이 필요로 한다[1], [4], [5], [14], [19]. 그러나 기존의 MIN에서는 모든 통신 쌍들간에는 스테이지 수와 동일한 거리가 항상 유지되기 때문에 지역 참조를 활용할 수 없게 된다. 이러한 지역 참조성의 손실은 시스템 성능을 저하시키는 한 가지 요인이 될 수 있다. 일반적으로 단일 프로세서 시스템 환경에 있어서는 대부분의 메모리 참조는 메모리 위치상에서 아주 적은 부분에서만 발생하게 된다. 이러한 연구는 캐쉬를 기반으로 한 시스템(cache based system)의 발전을 성공적으로 이루게 되었다. 이와 유사하게 다중 프로세서 시스템 환경

* 여수대학교 컴퓨터공학과 교수
** 상주대학교 소프트웨어공학과 부교수
논문번호 : 040127-0323, 접수번호 : 2004년 3월 23일
※ 본 논문은 정보통신연구진흥원 2002 대학 기초연구지원사업 연구비로 연구되었음.

하에서의 많은 대부분의 응용 프로그램에서는 프로세서간의 통신(interprocessor communication)은 주로 프로세서-메모리들의 작은 크기를 갖는 그룹에서 발생하게 된다[6],[7],[11]. 따라서 수많은 프로세서를 갖는 대형 시스템에서 각 프로세서, 메모리 쌍 간에 모두 동일한 길이의 연결 경로를 제공하기 보다는 통신이 자주 발생되는 작은 그룹에 더 빠른 경로를 제공함으로써 보다 향상된 시스템 성능을 얻을 수 있을 것이다. 이러한 프로세서들 간에서 통신 분포의 지역화를 본 논문에서는 지역 참조성이라는 표현으로 사용할 것이다. 아래의 예는 참고문헌 [1]을 기초로 하여 기존의 MIN에서 통신의 빈도수가 높은 그룹에 빠른 경로를 제공함으로써 얻게 되는 이점을 보인 것이다.

[예 1] 한 시스템에 4개의 프로세서(P1 ~ P4)가 있다고 하자. 그리고 이들 프로세서간의 통신 빈도수를 측정한 결과 [표 1]에서와 같이 산출되었다고 하자.

[표 1] 프로세서의 상호 통신 분포

Communication Between Processors (Total communication of each processor normalized to 1)				
	P1	P2	P3	P4
P1		0.7	0.2	0.1
P2	0.7		0.1	0.2
P3	0.2	0.1		0.7
P4	0.1	0.2	0.7	

각 프로세서 쌍에 따른 통신의 빈도수를 살펴보면, P1과 P2 그리고 P3과 P4가 각각 0.7로서 다른 프로세서 쌍들 보다 높게 나타난 것을 볼 수 있다. 만약 이들을 2개의 그룹 {P1,P2}와 {P3,P4}로 나누어 구성시킨다면, 이들에 대한 상호연결 네트워크를 MIN으로 구성할 때 적용되는 스테이지수는 $\log_2 p$개(여기서 p는 프로세서의 수)이므로 이들에 대한 평균 통신 지연은, $0.7\times\log_2 2+(0.2+0.1)\times\log_2 4=1.3$으로써, 그룹화 시키지 않았을 경우의 통신 지연, $\log_2 4=2$ 보다 적은 통신 지연 시간을 얻을 수 있다. 따라서 일반적으로 많이 그리고 자주 사용되는 응용 프로그램의 통신의 형태 등을 추적 도구(tracer tool)를 이용하여 이들에 대한 통신 분포를 알아낼 수 있다면, 이러한 프로세서-메모리 그룹에 보다 짧은 경로를 제공함으로서 시스템 성능을 보다 향상시킬 수 있을 것으로 기대할 수 있다.

그러나 기존의 MIN에서는 모든 프로세서-메모리 간의 통신 거리는 항상 n=log2N(N×N MIN에서 스테이지 수)이기 때문에, 프로세서의 수의 증가로 인해 스테이지 수가 증가되어 지연 시간이 점점 더 길어지게 된다. 따라서 이러한 수많은 프로세서를 갖는 대형 시스템에서 각 프로세서, 메모리 쌍 모두에 이렇게 동일한 길이의 연결 경로를 제공하기 보다는, 통신이 자주 발생되는 작은 그룹에 더 빠른 경로를 제공하여 통신의 지역 참조 성을 활용할 수 있는 MIN을 개발함으로서 보다 향상된 시스템 성능을 얻을 수 있을 것이다.

2. 관련 연구

Baseline, Omega등[8],[9],[10]의 기존의 MIN에서는 한 개의 근원지와 목적지간의 쌍(pair)에 대해서 단일 경로(UPP: Unique Path Property)[10]만을 제공하기 때문에 네트워크 상에서의 오류 및 트래픽에 대한 대체 경로가 전혀 없기 때문에 시스템 성능이 크게 저하되는 것은 자명한 사실이 된다. 따라서 이러한 단점을 보완하기 위해서, 즉 기존의 단일경로 MIN에서 다중경로를 제공하기 위해 중복 링크를 사용하거나 스테이지를 늘리거나 또는 네트워크의 다중 복사 본을 사용하여 중복 경로를 제공하는 것으로서 추가적인 하드웨어를 첨가함으로써 이루어진 연구만 진행되어 왔다[8],[17],[18]. 그러나 현재 및 미래에 사용될 수십-수천개 이상의 프로세서를 사용하는 MPP 시스템에서는 이렇게 추가되는 하드웨어 비용으로 인한 시스템의 가격의 상승 또한 신중히 고려되어야 할 요소일 것이다. 이에 부합하기 위해 본 논문에서는 하드웨어를 추가하여야만 중복 경로를 얻을 수 있다는 기존 연구[2],[4]와는 달리 기존의 MIN 보다 오히려 적은 하드웨어 비용만으로도 FAC(Full Access Capability)를 만족할 뿐만 아니라 지역 참조성을 활용할 수 있는 cost-effectiveness한 MIN을 새로이 제안하고자 한다.

또한 본 논문에서는 동적 상호연결 네트워크인 MIN에서 지역 참조성을 부여하기 위해서는 지역 참조성 활용에 대한 장점을 갖고 있는 정적 상호연결 네트워크의 위상을 고려하여, 이들 두 위상의 장점을 결합하여 새로운 동적 상호연결 네트워크 설계를 유도할 것이다. 지역 참조성 활용에 대한 장점을 갖고

있는 정적 상호연결 네트워크[5],[9],[12],[13] 중에서 hypercube[5],[9]는 네트워크의 크기의 증가(노드수의 증가)에 따른 추가적인 통신 포트(communication port)가 필요하게 된다. 따라서 hypercube의 노드에 대한 degree는 n으로서 네트워크의 크기에 비례하여 증가하게 되므로 hypercube의 위상(topology)은 진정한 확장성(scalability)을 갖고 있다고는 말할 수가 없다.

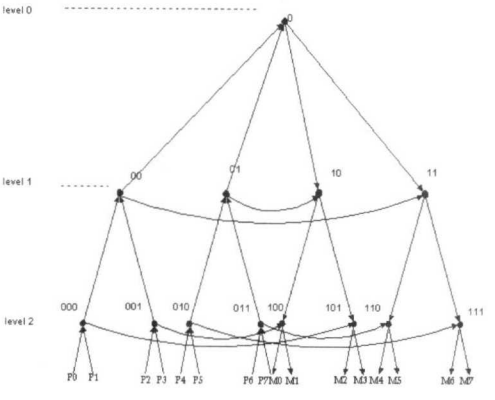
한편, 다른 정적 네트워크인 이진 트리(binary tree)구조[5],[9]에서는 degree가 3(루트 노드는 2)으로 고정적으로 일정하므로 확장성면에서는 좋은 상호연결 네트워크라 할 수 있다. 그리고 지역성이 높은 노드와는 거리가 1인 이웃노드로 연결될 수 있어, 지역 참조 역시 활용할 수 있다. 이러한 트리 네트워크구조는 병렬처리 데이타 베이스 및 여러 응용분야에서 많이 사용되고 있다. 그러나 원거리 통신에 있어서는 매우 긴 직경(diameter)을 갖게 되고, 또한 루트(root)노드에 대한 부하가 높게 나타날 수도 있지만, 이러한 단점은 추가적인 링크를 연결시킴으로써 이를 극복할 수 있는 연구가 Goodman[7]에 의해 이루어졌다.

동적 네트워크인 MIN은 쉬운 자기경로(self-routing)가 가능하며, 네트워크의 크기가 증가하여도 스위칭 소자 크기(degree)가 일정하게 유지될 뿐만 아니라 직경(diameter) 또한 $\log_2 N$으로서 n-cube의 직경과 동일하다. 그러나 이러한 직경은 프로세서와 메모리 모듈 쌍들 간에서 통신 지연은 항상 스테이지 갯수인 $\log_2 N$ 만큼 소요되기 때문에 국부 참조성을 활용할 수 없게 된다. 이는 통신이 국부참조의 통신이 많이 발생할 경우에서는 위에서 언급된 hypercube와 트리와 비교하였을 때 시스템 성능의 저하가 예상된다. 이러한 현상은 시스템의 크기가 커질수록 그 문제는 매우 심각하게 나타날 수 있다.

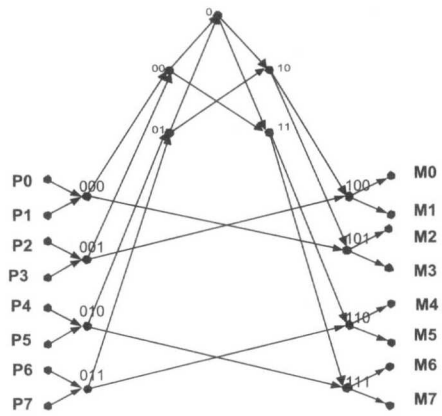
이렇게 hypercube 또는 tree와 MIN의 서로의 장점을 결합하기 위해 정적 네트워크와 동적 네트워크의 위상을 통합시키기 위한 방법이 필요하다. 따라서 이들 두가지 네트워크의 위상은 효율적으로 분석하기 위해 그래프 표현(graph representation) 방식[9],[10]을 이용할 것이다. 정적 네트워크 그래프 표현에서는 각 노드는 프로세서와 메모리 모듈을 포함한 표현이며, 각 링크는 그 링크와 연결된 노드간의 통신라인을 표현한 것이다. 동적 네트워크의 그래프 표현에서 링크의 표현은 정적 네트워크에서의 링크와 동일한 표현의 의미를 가지고 있지만, 각 노드는 스위칭 소자를 의미한다. 또한 그 네트워크의 왼쪽에는 프로세서들이 위치하고 오른쪽에는 메모리 모듈(프로세서)들이 위치하게 된다. 따라서 이들 두 그래프 모델을 일치시키기 위해서 먼저 이들 두가지 개념을 한가지로서 정립하여야 한다. 이렇게 정립된 통합 그래프 모델을 이용하여 일반적인 상호연결 네트워크 위상을 분석하여 적은 직경과 degree를 갖고, 국부 참조를 활용할 수 있는 상호연결 네트워크 설계 한다. 이러한 분석을 통해서 얻어진 통합 그래프 모델을 가지고 역으로 노드를 스위칭 소자로 그리고 링크를 스위치 소자와 연결된 라인으로 바꾸어서 동적 상호연결 네트워크 설계를 유도하여 우수한 성능을 갖는 새로운 부류의 상호연결 네트워크를 설계하고자 한다.

3. Combine MIN

Combine MIN에서는 기존의 MIN에서 지역 참조성 활용을 할 수 있는 전략으로서 위상 구조상 지역 참조성을 갖는 정적 네트워크인 이진 트리 구조를 동적 네크워크인 MIN의 위상에 결합시킨 새로운 위상을 갖는 상호연결 네트워크를 설계하게 된다. 지역 참조성을 활용할 수 있게 하여 통신 빈도수가 높은 지역 참조의 경우에는 보다 빠른 경로를 제공함으로써 자주 발생되는 통신에 대한 지연 시간을 줄일 수 있을 것이다. 프로세서와 메모리 모듈로 구성된 작은 크기의 클러스터에 높은 지역화 통신 분포를 가지며, 또한 통신 지연 시간이 네트워크의 조합 능력(combinatorial power)보다 더 중요하게 되는 병렬 응용 분야에 효과적으로 사용할 수 있는 상호연결 네트워크의 설계를 목표로 한다.



(a)



(b)

<그림 1> 프로세서-메모리 트리의 동일레벨에서의 노드 연결

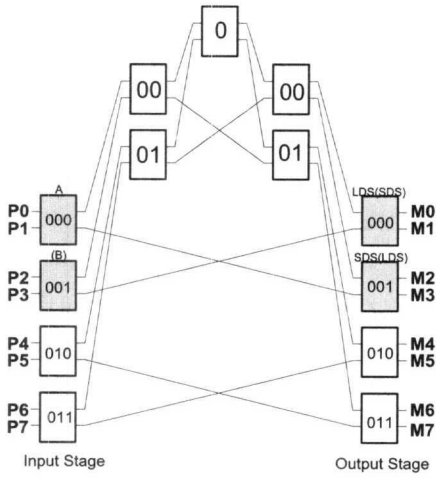
<그림 1>에서 (a)와 같은 tree의 모든 노드들은 각각 2-input, 2-output degree를 가지고 있기 때문에 2×2 스위칭 소자를 갖는 MIN을 형성할 수 있게 된다. <그림 1>의 (a)에서 tree의 아래 leaf 노드들 중에서 프로세서 노드들은 왼쪽 그리고 메모리 모듈들은 오른쪽으로 양쪽을 잡아 늘린다면, <그림 1>의 (b)와 같은 형태가 될 것이다. 이와 같이 늘린 형태의 그래프에서 각 노드를 2×2 스위칭 소자로 바꾸어 표현하면, <그림 2>와 같은 MIN을 형성시킬 수 있다. 이렇게 형성된 MIN을 8×8 Combine MIN이라고 하겠다.

Combine MIN은 한 프로세서당 2개의 메모리 모듈을 제외한 모든 메모리 모듈에 대해 중복 경로를 제공하고 있다. 이렇게 한 프로세서에서 연결 경로가 오직 한 개만 존재하는 2개의 메모리 모듈들은 다음과 같이 분석될 수 있다. 임의의 한 프로세서 $P_\ell P_{\ell-1}\ldots P_1P_0$와 직접 연결된 입력 스테이지의 스위칭 소자의 번호는 $P_\ell P_{\ell-1}\ldots P_1$와 같다. 입력 스테이지에 있는 이러한 스위칭 소자 $P_\ell P_{\ell-1}\ldots P_1$와 동일 번호를 갖는 마지막 출력 스테이지에 있는 스위칭 소자에 직접 연결된 2개의 메모리 모듈은 Combine MIN의 위상에 따라 $P_\ell P_{\ell-1}\ldots P_10$ 와 $P_\ell P_{\ell-1}\ldots P_11$의 번호를 갖는 메모리 모듈이 된다. 따라서 입력 스테이지에서의 스위칭 소자의 번호와 출력 스테이지에서의 스위칭 소자의 번호가 동일한 경우에는 그들과 연결된 프로세서-메모리 쌍들 간에는 오직 한 개의 유일 경로밖에는 존재하지 않는다. 따라서 이들 두 스위칭 소자를 장거리 스위칭 소자 (LDS: Long Distance SE) 관계라고 하겠다.

그리고 $P_\ell P_{\ell-1}\ldots P_1$의 Down 포트는 출력 스테이지에 있는 스위칭 소자, $P_\ell P_{\ell-1}\ldots \overline{P_1}$로 링크가 연결되어져 있다. 따라서 이들 두 스위칭 소자를 단거리 스위칭 소자(SDS: Shortcut Distance SE)관계라고 하겠다. 이러한 정의로 말미암아 주어진 한 프로세서에 대해 LDS 관계에 있는 스위칭 소자와 SDS 관계에 있는 스위칭 소자들은 Combine MIN에서로 바로 이웃하여 위치하고 있다. 또한 입력 스테이지에서 스위칭 소자, $P_\ell P_{\ell-1}\ldots P_1$와 바로 인접된 스위칭 소자의 번호는 $P_\ell P_{\ell-1}\ldots \overline{P_1}$ 이므로, 이러한 스위칭 소자 $P_\ell P_{\ell-1}\ldots \overline{P_1}$를 위와 같은 동일한 방법으로 LDS, SDS 관계를 적용시킨다면, 앞에서의 스위칭 소자, $P_\ell P_{\ell-1}\ldots P_1$에서와 정 반대적인 현상이 나타나게 된다.

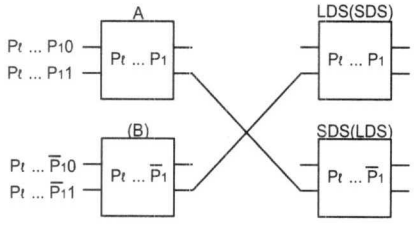
예를 들어 <그림 2>에서 입력 스테이지에 있는 0002의 번호를 갖는 스위칭 소자와 LDS 관계에 있는 스위칭 소자는 마지막 스테이지에서 번호 0002를 갖는 스위칭 소자이다. 이렇게 LDS 관계에 있는 스위칭 소자 0002에 연결에 직접 연결된 2개의 메모리 모듈들은 오직 한 개의 유일 경로만이 존재한다. 또한 입력 스테이지에 있는 0002의 번호를 갖는 스위칭 소자와 SDS관계에 있는 스위칭 소자는 출력 스테이지에 있는 스위칭 소자 0012이다. 따라서 이러한 스위칭 소자와 직접 연결된 메모리 모듈들은 2개의 경로를 가질 뿐만 아니라 최단 거리를 제공하게 된다. 또한 LDS 관계에 있는 스위칭 소자 0002와 SDS 관계에 있는 스위칭 소자 0012은 <그림 2>와 같이 서로 바로 이웃하여 위치하게 된다.

또한 입력 스테이지에서 스위칭 소자 0002와 바로 이웃한 스위칭 소자는 0012이다. 이 스위칭 소자에 위와 같은 방법으로 LDS, SDS 관계를 적용시키면, 앞에서 고려한 스위칭 소자 0002에서의 LDS 와 SDS 관계에 있는 출력 스테이지에 있는 스위칭 소자가 서로 정 반대로서 <그림 2>에서와 같이 각각 0012과 0002로 형성되어 진다.



<그림 2> Combine MIN에서의 LDS와 SDS

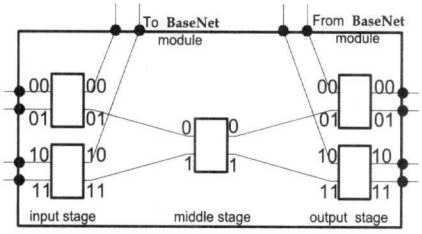
따라서 입력 스테이지에서 서로 인접된 스위칭 소자들의 LDS와 SDS 관계에 있는 출력 스테이지의 스위칭 소자들이 서로 정반대로서 서로 교차되어 있는 형태를 하고 있는 이러한 일반적인 관계를 그림으로 표현하면 <그림 3>와 같다.



<그림 3> LDS-SDS 쌍

<그림 3>에서와 같이 LDS와 SDS 관계가 서로 교차되어 있는 관계로서 이들간의 연결 링크들에서는 서로 교차점(cross point)이 발생하게 된다. 이러한 LDS, SDS의 교차점에 2×2 스위칭 소자를 추가하여 설치한다면 LDS 관계에 있는 스위칭 소자로의 새로운 경로를 만들어 낼 수 있어 앞에서 제안된 Combine MIN에 2개 이상의 추가적인 중복 경로를 제공할 수 있을 것이다.

이에 대한 실제 하드웨어에 구현할 수 있는 UNIT module은 <그림 4>와 같이 형성될 수 있다. 이것은 LDS, SDS 관계에서 구하여진 이러한 교차점에 2×2 스위칭 소자를 첨가한 형태가 되는 것이다. 따라서 LDS를 통하여 연결될 수 있는 메모리 모듈들에 대해서는 2개 이상의 중복 경로를 허용할 수 있게 된다. 그리고 이렇게 형성된 UNIT module을 기초로 한 점진적 확장(incremental scale)기법을 통하여 Combine MIN를 설계할 수 있다.



<그림 4> UNIT 모듈

4. Combine MIN에서의 라우팅

Combine MIN은 분산적 자기제어 라우팅(distributed self-routing) 전략을 이용하여 쉽고, 빠른 경로 선택을 지원할 수 있게 된다. Combine MIN은 위상적 구성상 UNIT module을 최소 단위의 설계 기본 module로 하고 있기 때문에 최소 길이를 갖는 경로는 이 UNIT module 내의 프로세서-메모리 쌍에 존재하게 된다.

따라서 한 프로세서에서 각각의 메모리까지의 연결을 위해서 3개의 스위칭 소자만을 통과하므로 전체 네트워크의 크기의 증가에 변화 없이 그 길이는 3으로서 항상 일정하게 유지될 수 있다. 따라서 이들 프로세서-메모리 그룹에서는 지역 참조 활용에 있어 매우 유리한 거리를 가질 수 있게 된다. 또한 모든 라우팅 경로를 지원할 수 있도록 하기 위하여 Combine MIN에서 라우팅에 사용하게 될 라우팅 태그의 총 비트 수는 2n-1개로써 아래와 같은 형태로 표현되어 진다.

t_{2n-2}	t_{2n-3}	t_{2n-4}	t_{2n-5}	.....	t_1	t_0

이러한 라우팅 태그의 각 비트 i(0≤i≤2n-2)는 Combine MIN의 각 스테이지 i에 있는 스위칭 소자의 내부 연결을 위한 제어 비트이다. 예를 들어, 스테이지 i에 있는 스위칭 소자의 입력 포트에 도착한 패킷의 라우팅 태그 비트 중에서 $t_i = 0$이면, 그 스위칭 소자의 출력 포트 중에서 Up 출력 포트에

그 패킷을 통과시키고, 그렇지 않고, 만약 $t_i = 1$이면, 그 스위칭 소자의 출력 포트 중에서 Down 출력 포트에 패킷을 전송시킨다.

그러나 항상 이러한 2n-1개의 라우팅 태그 비트를 모두 사용하지는 않는다. 일단 주어진 프로세서와 메모리에 대한 최단거리의 경로가 설정되면 그에 해당되는 라우팅 태그 비트만이 사용하게 된다. 그리고 그 경로가 블럭(block)되었거나 혼잡(busy)할 경우 1개의 계층씩 늘려서 재 라우팅(re-routing)을 할 수 있다. 이러한 재 라우팅은 라우팅 태그 비트를 2개씩 늘려 사용함으로써 그 경로를 얻을 수가 있다. 이렇게 라우팅 태그에 사용되는 비트 수가 계층에 따라 다르기 때문에 라우팅 태그 생성을 위해 주어진 한 프로세서에 대해 목적지 메모리 모듈을 포함하고 있는 계층을 결정하여야 한다. 이러한 계층별 라우팅 태그의 결정 방법은 n-1 가지의 계층에서 연결될 수 있는 메모리 모듈에 관한 이론을 기초로 하여 얻어진 방법으로 이루어질 수 있게 된다. 따라서 만약 근원지 주소가 $s_{n-1}s_{n-2}\ldots s_0$이고 목적지 주소가 $d_{n-1}d_{n-2}\ldots d_0$라고 하고 이들 두개의 비트-스트링(bit-string)중에서 부분 스트링, $s_{n-1}s_{n-2}\ldots s_2$ 와 $d_{n-1}d_{n-2}\ldots d_2$을 고려하여, 이들에 대해 exclusive-OR 연산을 적용시킨다.

따라서 그 결과로서 $c_{n-2}c_{n-3}\ldots c_1 = s_{n-1}s_{n-2}\ldots s_2 \oplus d_{n-1}d_{n-2}\ldots d_2$를 얻을 수 있다. 그리고 비트-스트링 $c_{n-2}c_{n-3}\ldots c_1$에서 MSB로 부터 시작하여 오른쪽 방향으로 LSB까지 스캔(scan)하여 최초의 1이 발견될 때까지 진행한다. 즉, C_i에서 최초의 1이 발견된다면, 이 메모리 모듈에 대한 계층은 i+1로 결정된다. 그러나 만약 그러한 C_i, 즉 모든 C_i가 0일 경우, 그 계층은 1로 결정된다.

위와 같은 라우팅 태그를 기초로 하여 계층 결정 방법을 이용한 라우팅 태그를 생성하는 알고리즘은 아래와 같다.

이와 같은 알고리즘으로 생성된 라우팅 태그를 이용하여 Combine MIN에서 분산적 자기 경로를 제공하게 된다. 이러한 라우팅 태그를 사용할 경우 스테이지 i에 있는 스위칭 소자로 라우팅 태그 t_{2n-2-i}를 검사하여 이것이 0이면, 그 스위칭 소자의 출력 포트 중 상단인 Up 포트로의 연결을 제공하고 만약 $t_{2n-2-i} = 1$이면, 그 스위칭 소자의 출력 포트 중 하단인 Down 포트로 입력된 패킷을 출력할 수 있도록 연결을 제공한다.

예를 들어 <그림 6>의 16×16 Combine MIN에서 근원지 010(0002)와 목적지 310(00112)사이를 연결할 수 있는 경로를 찾기 위해 선택될 수 있는 최단 경로를 근원지 비트-스트링 s_3s_2과 목적지 비트-스트링 d_3d_2에 각각 해당되는 근원지 비트-스트링 002과 목적지 비트-스트링 002을 exclusive-OR 연산을 수행하면, 그 결과 1을 갖는 비트를 찾을 수 없기 때문에 계층 1로서 결정된다. 따라서 라우팅 태그 생성 알고리즘에 의해 $t_{2n-2} = 1$로 세트되고, $t_1 = d_1$로, 그리고 $t_0 = d_0$로 할당되어 그 결과 라우팅 태그는 1112이 된다. 이에 대한 경로의 예는 <그림 6>의 점선으로 표시되어 있다. 그리고 다른 실선들은 우회할 수 있는 대체 경로를 표현하고 있다.

Routing Tag Generation Algorithm

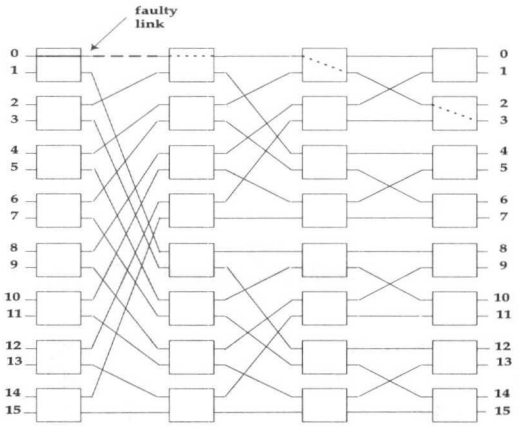
```
/*
  Src.  Address : sn-1sn-2 ... s0
  Dest. Address : dn-1dn-2 ... d0
*/

begin
 /* the decision of class */
    i = n;
    C = (sn-1sn-2 ... s2 ) ⊕ (dn-1dn-2 ... d2)
    while (ci≠1 and i≠0)  i=i-1;
    class = i;
 /* routing tag generation */
    j=2n-2;
    for(i=1 to class)
      begin
      tj = 0;
       j=j-1;
      end
      tj = 1;
    for(i=0 to class+1) ti = di;
 end.
```

5. 오류 허용 라우팅 전략

네트워크의 입력으로부터 요청된 패킷은 라우팅 태그를 사용하여 그 네트워크를 통과 시킬 경우에 그 경로상에서 교통 혼잡이 발생 하였거나 또는 링크 상에서의 오류가 발생 하였을 때, 그 패킷은 더

이상 다음 스테이지로 진행시킬 수 없게 된다. 이러한 경우에 그 스테이지에 해당되는 라우팅 태그 비트를 0에서 1로 교체함으로서 상단 계층을 통하여 우회하여 라우팅을 계속할 수 있게 된다. 본 연구에서 오류허용은 각 스테이지 i (0≤i≤n-1)에 있는 링크들에 대한 오류만으로 가정한다.

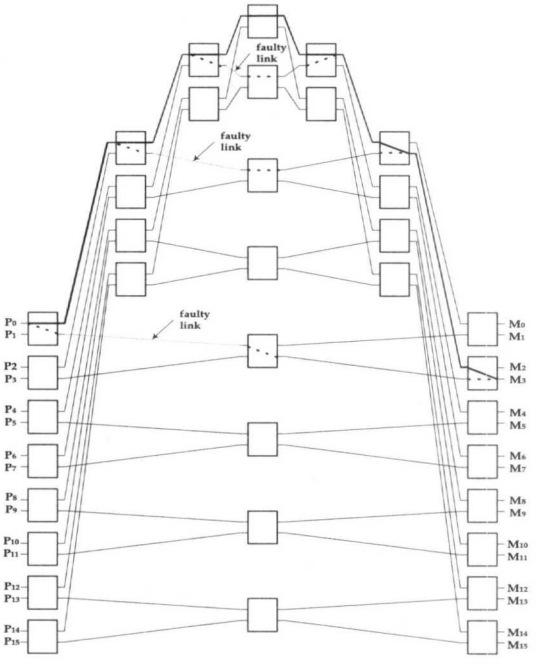


<그림 5> 16×16 MIN에서 링크 오류 발생시 대체 경로의 부재

제안된 Combine MIN은 기존의 MIN보다 스위칭 소자의 갯수 및 링크의 수가 훨씬 적은 수로써 최소한의 스위칭 소자로만 구성되었기 때문에 스위칭 소자에 대한 오류는 고려하지 않고, 오직 링크의 오류만을 적용시킨다. 이러한 링크만의 오류로 한정할지라도 기존의 UPP MIN에서는 전혀 다른 대체 경로를 제공하지 못하지만, 제안된 Combine MIN에서는 적은 수의 스위칭 소자 및 링크를 갖고 있을 지라도 많은 다른 대체 경로가 존재하게 된다.

이러한 방법으로써 한 개의 근원지와 목적지 쌍에 대한 최단 거리 라우팅 태그를 이용하여 라우팅하는 도중에 스테이지 i에서 Down 출력 포트에 많은 트래픽으로 인하여 정상적인 라우팅이 불가능해질 경우, 라우팅 태그 비트, t_{2n-2-i}를 0에서 1로 교체하고 $t_{n-2} = d_{n-2}$로 할당하여 새로운 대체 경로를 이용하여 라우팅을 계속하여 진행시킬 수 있다. 이러한 우회 대체 경로 방법은 계층이 n-1까지 계속 진행되고 또한 atom node에 이를 때까지도 진행할 수 있어 트래픽의 혼잡 및 링크의 오류가 발생될 때마다 우회할 수 있어 Combine MIN은 매우 간단하고 빠른 적응적 분산 자기제어 (adaptive distributed self-routing control)를 할 수 있게 된다.

<그림 5>와 <그림 6>은 네트워크의 크기가 16×16으로서 동일한 기존의 MIN (Baseline)과 Combine MIN에서 오류가 발생하였을 경우를 각각 설명한 것이다. <그림 5>의 기존의 MIN에서는 단지 한 개의 링크 오류가 발생하여도 이 링크를 통과하여야 하는 근원지 0과 목적지 3과의 통신은 다른 대체 경로가 존재하지 않아 연결이 영원히 불가능하게 되었지만, <그림 6>에서의 Combine MIN에서는 3개의 링크의 오류가 발생하여도 근원지 0과 목적지 3과의 연결 경로가 존재하여 이들을 연결시킬 수 있게 된다.



<그림 6> Combine MIN에서 링크 오류에 대한 적응적 라우팅 전략

<그림 6>에서는 앞에서 소개된 라우팅 태그 생성 알고리즘과 오류허용을 위한 적응적 라우팅 태그 교체 방법을 이용하여 새로운 경로를 결정하는 것을 보인 것이다. 여기서 점선은 오류가 발생된 링크를 나타내며, 만약 라우팅을 수행하는 도중 이러한 오류가 발생된 링크를 만나게 되면 위에서 설명되었던 오류 허용을 위한 적응적 라우팅 태그의 변환 전략을 통하여 이러한 오류 링크들을 우회하여 원하는 목적지까지의 연결을 이룰 수 있다.

여기서 관심 있게 살펴 볼 수 있는 것은 Combine MIN은 이러한 다중 경로를 가지고 있음에도 불구하

고 과거의 연구와 같은 추가적인 스위칭 소자를 사용하지 않고 오히려 기존의 MIN의 스위칭 소자의 개수 보다 훨씬 적은 수로써 이러한 중복 경로를 제공할 수 있는 것이다. [표 2]는 각 네트워크에서 사용되는 스위칭 소자의 개수와 중복 경로의 수를 비교한 것이다.

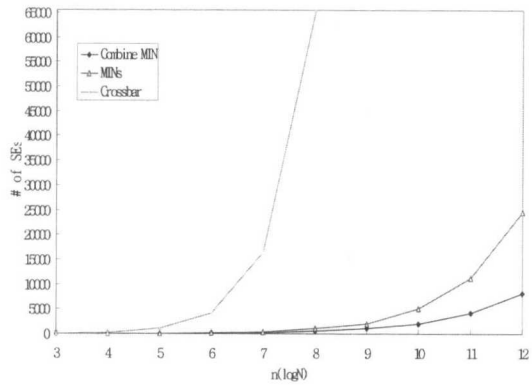
여기에서도 잘 나타나 있듯이 네트워크의 크기가 증가할수록 Combine MIN의 평균 경로 수는 증가하게 되고, 네트워크에서 사용되는 스위칭 소자의 수 또한 차이가 더욱 커지게 된다. <그림 7>은 [표 2]에서의 증가되는 스위칭 소자의 수를 그래프로 나타낸 것이다. 또한 Combine MIN의 하드웨어 복잡 도를 계산하기 위해서는 네트워크의 대부분을 차지하고 있는 스위칭 소자의 개수를 기준으로 산출하였을 때, Combine MIN에서 사용되는 스위칭 소자의 총 개수는 아래의 식과 같이 구할 수 있다.

$$
\begin{aligned}
&2N-3+2^{n-1}-1 \\
&=2\cdot 2^{n}-3+2^{-1}2^{n}-1 \\
&=2.5N-4
\end{aligned}
$$

따라서 Combine MIN에서 스위칭 소자의 수는 계속해서 O(N)을 유지할 수 있고, 또한 기존의 MIN에서의 스위치 소자의 하드웨어 비용, O(NlogN)에 비해 훨씬 적은 비용으로써 FAC를 만족할 뿐만 아니라, 네트워크의 크기의 증가에 중복 경로의 수는 증가하고, 또한 계층이 낮을수록 많은 중복 경로를 제공할 수 있어 최대 n개의 중복 경로를 제공할 수 있다. 또한 이들에 대해서는 짧은 경로를 제공할 수 있게 된다.

[표 2] 기존의 MIN과 Combine MIN에서 사용되는 스위칭 소자의 수와 제공될 수 있는 평균경로 수의 비교

Network Size(n)	사용된 스위칭 소자의 갯수		평균 경로 수	
	MIN	Combine MIN	MIN	Combine MIN
2	4	6	1	2
3	12	16	1	2.5
4	32	36	1	3
5	80	76	1	3.5
6	192	156	1	4
7	448	316	1	4.5
8	1024	630	1	5
9	2304	1276	1	5.5
10	5120	2556	1	6



<그림 7> 시스템 크기에 따른 사용되어지는 스위칭 소자의 수

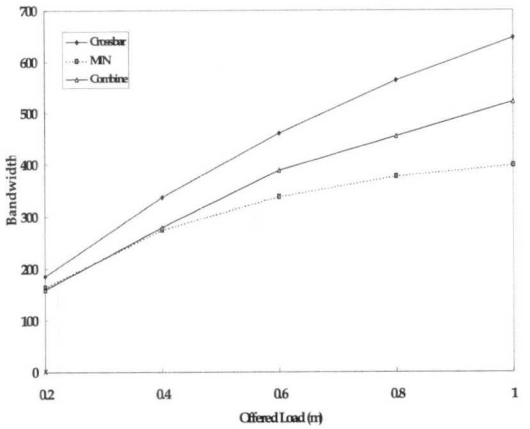
6. 시뮬레이션을 통한 성능 분석

본 절에서는 Combine MIN에 대한 성능을 평가하도록 하겠다. 기존의 MIN은 임의의 근원지에서 목적지간의 경로에 대한 거리가 고정되어 있어, 항상 두 쌍간의 거리는 n으로 일정한 반면, Combine MIN에서는 통신의 지역 참조성의 정도에 따라 그 거리는 달라지게 된다. 이때, 각 프로세서는 참조되는 메모리 모듈에 대한 거리는 2에서 부터 2n-1까지 다양한 길이를 가질 수 있다. 다시 말해서, 프로세서와 메모리 모듈사이에서 통신의 빈도수가 높은 쌍들에 대해서는 메모리 액세스 시간을 줄이기 위해 짧은 거리의 링크를 부여하고, 통신 빈도수가 낮은 메모리 모듈들은 보다 통신 거리가 긴 링크를 부여하도록 하여 메모리 액세스 시간의 차이를 두는 것이다.

본 절에서는 제안된 네트워크를 실제로 구현하기에 앞서 컴퓨터를 이용한 시뮬레이션 모델을 통하여 수학적인 분석 방법을 통한 성능 분석을 검증하고, 또한 실제 시스템의 동작과 관련된 다양한 환경 혹은 시스템 내에서 존재할 수 있는 여러 변수 요인들의 변화에 의하여 발생하는 영향을 시뮬레이션 모델에 적용시켜서 이에 대한 결과를 연구 분석하고, 제안된 새로운 시스템의 성능을 예측하기 위한 도구로서 컴퓨터 시뮬레이션을 수행하였다. 본 논문에서의 상호연결 네트워크는 Scientific and Engineering Software Inc.사의 SES/workbench[17]를 사용하여 이산 사건 모델링(discrete event modeling)으로 시뮬레이션을 수행하였다.

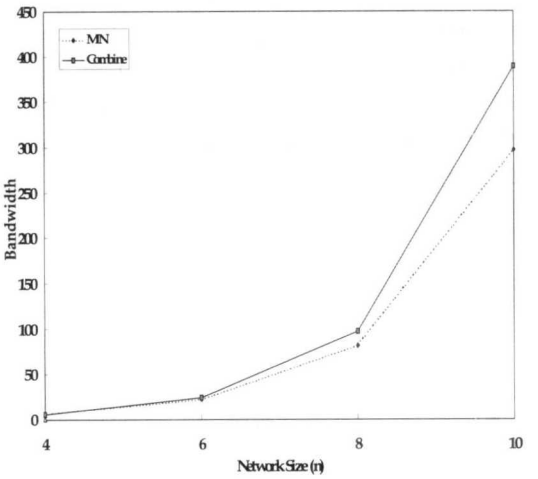
본 논문에서의 시뮬레이션의 형태는 기존의 MIN

과 제안된 Combine MIN에 대한 네트워크 시스템의 구성 요소 및 이들에 대한 동작의 변화 등을 모델링 함으로써 이루어졌다. 모델링은 실제 시스템의 동작과 관련된 가정들을 선정하여 수행하였다. 따라서 앞 절에서 분석하였던 예측된 분석결과와 시뮬레이션을 통한 실험 결과를 서로 비교하여 검증 평가하여, 실제 시스템을 제작하기 전에 정확한 시스템을 근접하게 추정할 수 있는 분석 도구로 사용할 수 있도록 한다.



<그림 8> 네트워크의 부하 변화에 따른 BW

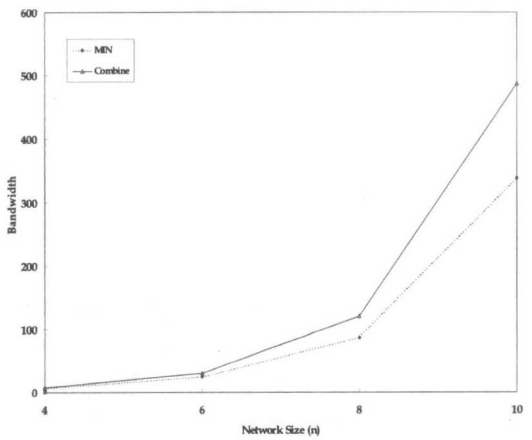
성능 분석 결과로서 <그림 8>에서는 네트워크의 크기가 1024×1024인 3가지 네트워크들을 부가된 부하의 변화에 따라 BW에 대한 비교를 나타낸 것이다. 이것은 동일한 1024×1024의 네트워크 크기를 갖는 3가지 네트워크, 즉 크로스바 스위치 네트워크와 기존의 MIN, 그리고 제안된 Combine MIN에서 가장 높은 지역화의 통신 분포를 갖는 환경에서 네트워크의 입력을 통해서 들어오는 요청의 비율을 점차 증가시켰을 때, 각 네트워크가 나타내는 성능을 서로 비교 평가하기 위한 것이다. Combine MIN은 시뮬레이션 및 분석 결과 모두가 MIN의 대역폭(BW : bandwidth)보다 최대 50%이상의 향상을 나타내고 있다. 이러한 성능의 향상은 Combine MIN은 위상 구조상 높은 지역화 통신 분포를 갖는 클러스터에 짧은 경로를 제공할 뿐만 아니라 이들에 대해서는 많은 수의 중복경로를 갖고 있기 때문에 나타나는 것을 볼 수 있다. 그리고 Combine MIN에서 분석 결과가 시뮬레이션 결과 보다 BW가 좋게 나타나는 것은 분석적 방법이 보다 이상적인 통신 형태가 적용되었기 때문으로 판단된다.



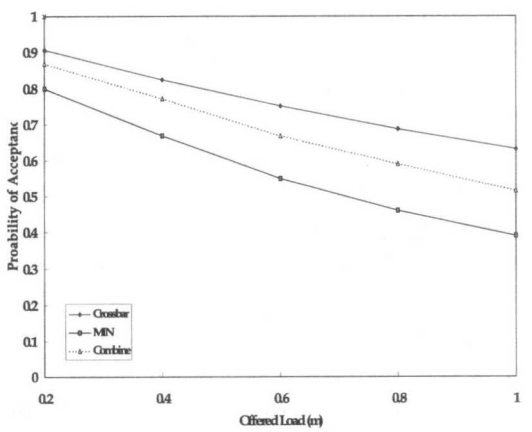
<그림 9> 지역 참조성이 0.6일때 네트워크 크기에 따른 BW

<그림 9>에서 <그림 10>은 Combine MIN의 네트워크 크기를 변화시키면서 지역 참조 율의 변화에 대해 BW를 분석한 실험 결과이다. 이것은 지역 참조율의 변화가 기존의 MIN과 Combine MIN의 성능에 어느 정도의 영향을 미치는지를 분석하기 위한 것이다. <그림 11>은 1024×1024 크기를 갖는 네트워크들에 부하를 증가시키면서 시뮬레이션과 분석 방법을 통해서 PA(Probability of Acceptance)를 비교한 것이다. 이러한 실험은 주어진 네트워크에 부하를 최대로 증가시켰을 경우 PA가 어느 정도까지 유지할 수 있을 것인가를 평가하기 위한 분석이다. Combine MIN은 시뮬레이션과 분석 방법 모두가 기존의 MIN 보다 PA가 최대 0.3 정도 향상됨을 볼 수 있었다. 비록 부하가 증가할 때는 제안된 Combine MIN의 PA도 감소하지만, 기존의 MIN과 비교하였을 때는 Combine MIN이 높은 PA가 나타나는 결과를 얻을 수 있었다. 이러한 결과는 Combine MIN의 대체 경로수가 많이 존재하기 때문에 요청이 네트워크를 통해서 요청이 연결될 수 있는 확률이 높기 때문이다. <그림 12>는 1024×1024 크기를 갖는 기존의 MIN과 Combine MIN에서 전체 네트워크의 링크 중, 2개에서 최대 32개까지의 링크 오류를 발생시켰을 때 지역 참조율의 변화에 따른 PA를 평가한 것이다. 이것은 네트워크 상에서 동일한 수의 오류가 동일한 위치에 오류가 각각 발생하였을 경우 어느 정도의 요청을 네트워크에서 받

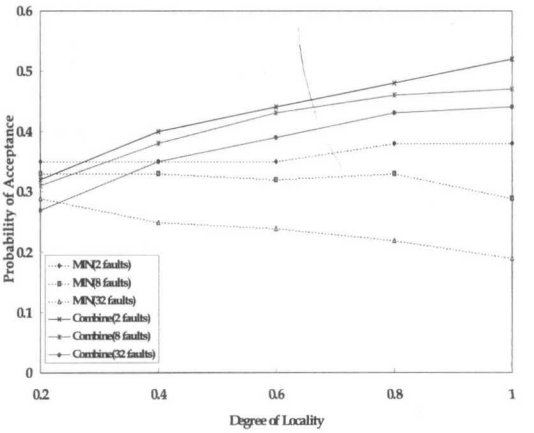
아들여 처리할 수 있는지를 분석하기 위한 실험이다. Combine MIN에서는 2개 이상의 중복 경로가 존재하므로 오류의 개수가 증가하여도 PA의 감소는 기존의 MIN 보다 훨씬 작음을 볼 수 있다. 그러나 기존의 MIN은 이러한 오류 수의 증가에 대해 감소 폭이 클 뿐만 아니라, 단일 경로를 갖고 있기 때문에 경로가 존재하게 되어 PA의 감소율이 보다 적은 것으로 나타나고 있다.



<그림 10> 지역 참조성이 0.8일때 네트워크 크기에 따른 BW



<그림 11> 1024×1024 Combine MIN에서 부하 변화에 따른 PA



<그림 12> 지역 참조율의 변화와 링크 오류 증가에 따른 PA

7. 결 론

본 연구에서 제안된 Combine MIN은 정적 네트워크인 이진 트리 위상의 장점과 동적 위상인 MIN의 장점을 결합함으로서 지역 참조성의 활용과 적은 수의 스위칭 소자로써 대체 경로를 제공하고, 목적지 주소를 이용한 간편한 분산적 자기경로 제어 라우팅을 적용시킬 수 있을 뿐만 아니라 또한 계층 버스에서와 같은 트래픽의 고립화(isolation)를 MIN에서도 적용시킬 수 있어 통신의 효율성을 높일 수 있게 되었다. 또한 Combine MIN이 기존의 MIN에서 사용되는 적은 수의 스위칭 소자를 사용함에도 불구하고 임의의 한 프로세서에서 모든 메모리 모듈로 접근할 수 있으며 지역화 된 클러스터에 다중 경로를 존재하여 시스템 크기가 증가할수록, 네트워크 상에서 오류 허용 및 교통의 혼잡에 대해 이를 우회할 수 있게 하는 다중 경로의 수 또한 증가함을 보였다. 따라서 Combine MIN은 하드웨어 비용과 성능 면을 고려하였을 때 기존의 MIN보다 효율적인 네트워크이며, 또한 적은 수의 프로세서와 메모리로 구성된 클러스터 내에서 통신의 빈도가 높게 지역화된 MPP 시스템 환경 하에서 동일 크기의 기존의 MIN 보다 우수한 상호연결 네트워크 구조이어서 지역화 된 통신 형태에서 뿐만 아니라 균일 분포를 갖는 병렬 프로그램 응용 분야에서도 우수한 성능을 나타낼 수 있는 MIN을 설계할 수 있을 것으로 기대된다.

참 고 문 헌

1. S.G. Abraham, and E.S. Davidson, "A Communication Model for Optimizing Hierarchical Multiprocessor System," In Proc. Int'l Conf on Parallel Processing, pp.467-474, 1986.
2. G.B. Adams III, D.P. Agrawal, and H.J. Siegel, "A Survey and Comparison of Fault - Tolerant Multistage Interconnection Network," IEEE, Compt., pp.14-27, June, 1987.
3. R. Agrawal and H.V. Jagadish, "Partitioning Techniques for Large-Grained Parallelism," IEEE Trans. Compt., vol. C-37, pp.1627-1634,. Dec., 1988.
4. L.N. Bhuyan and D.P. Agrawal, "Performance of Multiprocessor Interconnection Networks," IEEE Compt., pp.25-37, Feb., 1989.
5. A.L. Decegama, The Technology of Parallel Processing : Parallel Processing Architectures and VLSI hardware volume I, Prentice-Hall International Editions, 1989.
6. M. Dubois and S.S. Thakkar, Cache and Interconnect Architectures in Multiprocessors, Kluwer Academic Pub., 1990.
7. J.R. Goodman, "Using Cache Memory to Reduce Processor-Memory Traffic," Proc. 10th Symp., Computer Arch., pp. 124-131, June, 1983.
8. H.B. Hadin, "The Multi-Level Communication: Efficient Roting for Interconnection Networks, The Journal of Supercomputing, vol.18, no.2, 2001.
9. K. Hwang, Advanced Computer Architecture: Parallelism Scalability Programmability, McGraw-Hill International Editions, 1993.
10. K. Hwang and F.A. Briggs, Computer Architecture and Parallel Processing, McGraw-Hill Inc., 1984.
11. S.C. Kothari, "Multistage Interconnection Networks for Multiprocessor Systems," Advances In Compt,. vol. 26, 1987.
12. L.L.Ling, A. Fiho, " D-ARM : A New Proposal for Multi-dimensional Interconnection Network", ACM SIG COMM., pp.33-58, 2001.
13. Y.Li, S. Peng, and W. Chu, " Metacube: A New Interconnection Network for Large Scale Parallel System", Proc. of the Seventh Asia-Pacific Conference on Computer Architecture vol.6, pp.29-36, 2002
14. N. Suzuki, Shared Memory Multiprocessing, The MIT Press, 1992.
15. D.A. Patterson and J.L. Hennessy, Computer Architecture A Quantitative Approach, Morgan Kaufmann Pub., 1996.
16. D.K. Pradhan, Fault-Tolerant Computer System Design, Prentice-Hall PTR, 1996.
17. SES/workbench Rel. 3.0, Scientific and Engineering Software, Inc., 1995
18. H.J. Siegel, Interconnection Networks for Large-scale Parallel Processing, Lexington books, 1985.
19. H. Wand, "CPMBIC: An improved Cluster-based interconnection network", Int'l Journal of Computer Applications in Technology, vol.13, pp.3-5, 2000.

장 창 수(Chang-soo, Jang) 정회원

1980년 2월 : 조선대학교 공과대학 전자공학과 (학사)
1982년 8월 : 건국대학교 대학원 전자공학과 (공학석사)
1997년 2월 : 서강대학교 대학원 전자계산학과 (공학박사)
1999년 1월 ~ 2000년 2월 : 교환교수(California State olytechnic University)
1984년~현재 : 국립 여수대학교 IT학부 컴퓨터공학과 교수
<주관심 분야> : 병렬처리구조, 컴퓨터구조, DSP, 컴퓨터네트워크

최 창 훈(Chang-hoon, Choi) 정회원

1988년 2월 : 명지대학교 전자계산학과(학사)
1990년 2월 : 서강대학교 대학원 전자계산 학과(공학석사)
1997년 8월 : 서강대학교 대학원 전자계산학과 (공학박사)
1990년 : 대우통신(주) 기술개발부
1995년~1996년 : 미국 AT&T(NCR) 기술파견 연구원
1997년 9월~현재 : 국립상주대학교 컴퓨터공학부 전임강사 / 부교수

<주관심 분야> : 컴퓨터구조, 병렬처리시스템, 컴퓨터시뮬레이션