論文96-33B-5-6

상용 응용을 위한 병렬처리 구조 설계

(Design of the New Parallel Processing Architecture for Commercial Applications)

韓 宇 宗 *, 尹 碩 漢 *, 林 基 郁 *

(Woo-Jong Hahn, Suk-Han Yoon, and Kee-Wook Rim)

요 약

본 논문에서는 대규모 병렬처리 시스템의 확장성과 다중프로세서 시스템의 특성을 접목시킨, 클러스터 방식을 이용한 병렬처리 시스템을 제안하고 있다. 최근 고성능 마이크로 프로세서들의 가격 경쟁력에 힘입어 병렬처리 시스템들이 많이 등장하고 있으며, 이들 병렬처리 시스템들은 우수한 확장성을 갖고 있으나 주로 데이터 병렬성이 매우 강한 과학 연산 응용에 사용되고 있다. 상용 응용들은 주로 다중프로세서 시스템들을 서버로 이용하고 있으나 이들은 확장성에 한계가 있다. 본 논문에서 제안된 구조는 상용 응용을 대상으로 하면서 병렬처리 시스템의 우수한 확장성 및 성능을 제공하여 기존 다중프로세서 서버들의 한계를 극복할 수 있다. 제안된 시스템의 구조와 특징들을 분석하고 고유 상호연결망 구조를 분석한다. 고유 상호연결망은 계층 크로스바 연결망 구조를 갖고 있으며 클러스터 연결 방식에 적합한 프로토콜, 경로제어 방식, 신호전달 방식들을 제공한다. 상호연결망 설계 변수들을 분석하고 시뮬레이션을 통하여 제안된 구조가 목표 시스템에 적합한 특성을 유지하고 있음을 보이고 있다.

Abstract

In this paper, a new parallel processing system based on a cluster architecture which provides scalability of a parallel processing system while maintains shared memory multiprocessor characteristics is proposed. In recent days low cost, high performance microprocessors have led to construction of large scale parallel processing systems. Such parallel processing systems provides large scalability but are mainly used for scientific applications which have large data parallelism. A shared memory multiprocessor system like TICOM is currently used as a server for the commercial application, however, the shared memory multiprocessor system is known to have very limited scalability. The proposed architecture can support scalability and performance of the parallel processing system while it provides adaptability for the commercial application, hence it can overcome the limitation of the shared memory multiprocessor. The architecture and characteristics of the proposed system shall be described. A proprietary hierarchical crossbar network is designed for this system, of which the protocol, routing and switching technique and the signal transfer technique are optimized for the proposed architecture. The design trade-offs for the network are described in this paper and with simulation using the SES/workbench, it is explored that the network fits to the proposed architecture.

* 正會員, 韓國電子通信研究所
(Electronics and Telecommunications Research Institute)

※ 본 연구는 고속병렬컴퓨터 개발 사업의 일부로이루어졌음

接受日字:1995年4月27日, 수정완료일:1996年4月15日

Ⅰ. 서 론

ENIAC이 개발된 이래 컴퓨터 기술은 비약적인 발전을 거듭하여 최근에는 병렬처리 시스템들이 급속히 확산되고 있다. 병렬처리 구조란 충분히 큰 문제를 빠른 시간 내에 해결하기 위하여 동시 수행 특성을 효과적으로 이용할 수 있도록 고안된 구조라고 생각되어 왔다. 1980년대부터 많이 등장하기 시작한 초기 병렬처리 시스템들은 주로 연산용 서버, 즉 값싼 수퍼 컴퓨터라는 개념을 가지고 있었다. 그러나 그 시장 크기가 제한되어 있으므로 매우 큰 시장을 가지고 있는 상용 응용(commercial application)을 대상으로 하는 시장에 진출하려는 욕구가 있고, 상용 응용을 원하는 사용자들도 점점 강력한 시스템을 요구함에 따라 상용 병렬처리 시스템들이 등장하고 있다. 이들은 주로 1990년대부터 두각을 나타내기 시작하였으며 병렬 데이터베이스 엔진의 확산에 힘입어 매우 주목받고 있다. 병렬처리 시스템은 초기에는 대학과 모험 기업들에 의해 주도되었으나 최근에는 시장을 주도하는 대형 업체들도 여러 시스템을 발표하고 있으며, Cray Research사는 T3D, T3E를 IBM은 SP-1, SP-2들을 발표하고 있다.

이와 같은 병렬처리 시스템의 확산은 여러 관련 기술의 발전에 힘입은 것이다. 첫째, 상용 마이크로 프로세서들의 급속한 발전에 따른 것이다. RISC형 마이크로 프로세서가 등장한 이 후, 최근에는 200 MHz이상 되는 동작 속도를 갖는 마이크로 프로세서들이 발표되고 있다[1]. 이들은 이미 Cray-1의 성능을 넘어서고 있으며 2000년경에는 500 MHz에 달하는 동작 속도를 제공할 수 있을 것이다. 이들은 병렬처리 시스템의 가격 대비 성능을 비약적으로 향상시켰으며 고도의 프로세서 설계 기술을 보유하지 않더라도 병렬처리 시스템을 구현할 수 있게 되었다. 둘째, 높은 대역폭과 효율적인 프로토콜을 제공하는 상호연결망 기술 발전에 힘입은 것이다. 마이크로 프로세서 경우와 마찬가지로 반도체 기술의 발전에 따라 다양한 형태를 갖는 상호연결망들이 VLSI로 구현될 수 있으며 동작 속도도 매우 빨라지고 있다. 특히 CMOS 기술 발전이 지속됨으로써 고속 상호연결망 기술이 일부 업체 범위를 넘어 널리 확산되고 있다. 즉, 계속되는 회로 설계 및 공정상의 발전에 의하여 100 만개 이상의 transistor를 집적하고도100 MHz이상의 속도를 내는 CMOS VLSI들이 속속 발표되고 있고, 마침내 IBM, Unysis들의 전통적 업체들도 bipolar 기술 대신 CMOS를 대폭 채택하고 있다[2]. 프로토콜 측면에서도 사용 효율이 높은 스위칭 방법, 경로제어 방법들이 등장하여 병렬처리 시스템의 확장성을 높여 주고 있다. 세째, 소프트웨어 발전도 병렬처리 시스템 확산에 크게 기여하고 있다. 시스템 소프트웨어에서는 마이크로 커널 기술의 발전으로 사용자 편의를 해치지 않고 더욱 많은 프로세서들을 지원할 수 있게 되었다. 특히 Oracle Parallel Server(OPS)와 같은 병렬처리 구조를 지원하는 데이터베이스 엔진이 등장함으로써 많은 데이터베이스 응용들이 병렬처리 시스템에서 운용될 수 있게 되었다[3].

본 논문에서는 병렬처리 시스템의 장점을 제공하면서 상용 응용에 쉽게 접근할 수 있는 클러스터 기반의 고속 병렬처리 시스템을 제안하고 있다. 응용 프로그램을 위한 프로세서들을 256개까지 지원하는 병렬처리 시스템이면서 전체 시스템을 상용 응용 프로그램이 용이하게 운용될 수 있도록 여러 클러스터로 구분하고 있다. 각 클러스터는 데이터베이스 서버로서도 충분한 성능을 제공하며 처리해야 될 일이 더욱 많을 경우에는 한 시스템 이미지를 유지하면서 확장이 가능하다. 클러스터 내 연결망과 클러스터간 연결망은 같은 프로토콜을 사용함으로써 대역폭 사용 효율을 높이고 지연시간을 줄이고 있다.

본 논문은 다섯 장으로 구성되어 있으며 서론에 이어 2 장에서 병렬처리 시스템의 구조와 본 구조를 제안하게 된 배경을 기술하고 있다. 3 장에서는 본 논문에서 제안된 구조를 갖는 고속 병렬처리 시스템에 대하여 기술한다. 계층 크로스바 연결망 구조와 성능 평가를 4 장에서 기술하며 마지막으로 5 장에서 결론을 맺는다.

Ⅱ. 병렬처리 시스템 구조

많은 프로세서와 메모리를 갖는 병렬처리 시스템은 메모리 구성과 통신 방법에 따라 GMSV(Global Memory Shared Variable), DMSV(Distributed Memory Shared Variable), GMMP(Global Memory Message Passing) 그리고 DMMP(Distributed Memory Message Passing)으로 구분할 수 있다[4]. GMSV, GMMP와 DMSV, DMMP는 메모리 구성에 따라 나뉘어지며 모든 메모리를 프로세서들

이 공유하는 구조가 GMSV와 GMMP이다. GMSV와 DMSV는 서로 다른 메모리 구성이지만 모두 데이터 공유 방식으로 프로세서간 통신이 이루어진다. 이 방식은 프로세서간 통신 관점에서 단일 프로세서 시스템과 유사한 모델을 제공하므로 프로그래밍이 용이하다는 장점이 있다. 그러나 프로세서 수가 증가함에 따라 메모리에 대한 통신량이 급격히 증가하므로 확장성이 제한된다. GMSV는 Symmetry 시스템이나 TICOM들과 같이 프로세서 수가 수십 개 이내일 때 주로 사용된다[5] [6] [7]. DMSV는 확장성을 유지하면서 GMGV가 갖는 프로그래밍 모델을 제공하고자 제안된 것으로 최근에 활발히 연구되고 있는 DSM(Distributed Shared Memory) 구조가 이 종류에 속한다. 그러나 이 방식은 현재까지는 메모리 일관성 문제로 인하여 연결망 통신 부하가 매우 크므로 원래 목적과 달리 확장성이 제한된다. 데이터 공유 방식을 지원하는 시스템들로는 CMU의 Cmmp로부터 NYU의 Ultracomputer, IBM RP3, Stanford DASH, 그리고 KSR-1들이 있다[8] [9].

DMMP와 GMMP는 메시지 전송을 이용하여 프로세서간에 필요한 정보 교환 및 동기를 유지하는 방식이다. 프로그래밍 모델이 단일 프로세서 시스템과 다르다는 단점이 있으나 확장성이 매우 우수한 방식이다. 이 방식에서는 메모리 일관성 유지를 위한 통신 부담이 매우 적고 연결망을 통한 데이터 이동 횟수를 최적화하기 용이하다. GMMP는 예외적인 경우이다. 프로그램 개발을 쉽게 하고 초기에 이식을 돕기 위하여 소프트웨어에 의한 SVM(Shared Virtual Memory) 개념을 지원하기도 한다. DMMP 방식을 지원하는 시스템들로는 CosmicCube를 비롯하여 nCUBE사의 n CUBE 시리즈, Intel iPSC 시리즈와 Paragon, 그리고 Cray Research의 T3D, T3E들이 있다[10] [11].

대규모 DMMP 시스템들은 주로 연산 서버(computing server)로서 사용되고 있다. 연산 서버는 충분히 큰 문제를 해결하기 위하여 속도 향상(speed-up)을 주로 고려하여 설계된 것이며 다음과 같은 특성에 기초하고 있다. 첫째, 데이터 양이 많고 데이터 병렬성이 강한 응용을 대상으로 한다. 이것은 초기에 데이터 분배를 마치면 일정 기간 동안 프로세서간 데이터 교환이 거의 없고 프로세서들이 독립 운영되며, 통신이 필요한 시기도 미리 정의되므로 연결망 전체에 걸쳐 동기화가 용이하다. 또 문제 크기가 충분히 크므로 많은 작은 문제들을 동시에 처리해야 하는 일반 상용 응용에 비하여 데이터 지역성 문제가 덜하고 확장성을 유지하기가 용이하다. 둘째, 일반 상용 응용에 비하여 매우 한정된 알고리즘들을 대상으로 하고 있다. 따라서 각 구조에 맞게 프로그램을 최적화하는 것이 비교적 용이하다. 일반 상용 응용들은 워크스테이션들에서 볼 수 있는 바와 같이 그 범위가 매우 넓으므로 모든 경우 마다 모든 프로그램을 최적이 되도록 변경할 수는 없다.

위의 특성들은 OLTP(On-Line Transaction Processing), 데이터베이스 응용들에서는 살리기 어려운 것들이며, 이러한 이유로 데이터베이스 응용들에서는 비교적 작은 규모를 갖는 공유 메모리 다중프로세서 시스템들이 서버로 많이 사용되고 있다. 이들 응용에서도 데이터 양이 계속 늘어나고 서비스 종류도 늘어나고 있으며 특히 멀티미디어 데이터와 같이 많은 데이터를 실시간으로 처리해야 하는 요구가 늘고 있으므로 작은 규모의 다중프로세서 시스템보다 강력한 서버가 필요하다. 그림 1에 나타난 것과 같이1997년경에는 대규모 병렬처리 시스템들도 대부분 상용 응용을 지원할 것이라는 분석도 있다[12].

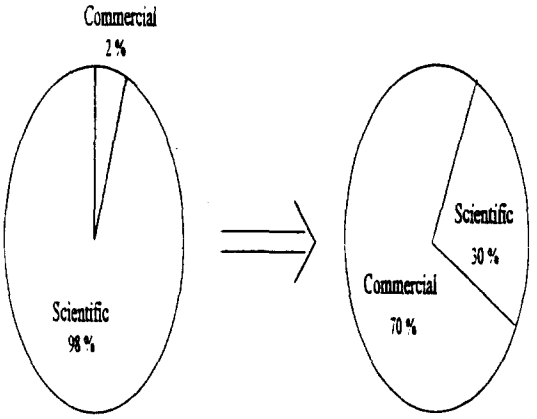


그림 1. 대규모 병렬처리 시스템 동향
Fig. 1. Massively Parallel Processing System Trend.

본 논문에서는 이와 같은 기술 및 시장 동향들과 현실적인 기술 수준들을 고려하여 OLTP 응용을 주요 대상으로 하는 클러스터 기반의 병렬처리 시스템 구조를 제안하고 있다. 작은 규모를 갖는 공유메모리 다중프로세서들을 여러 개 연결하여 한 클러스터를 구성하고, 클러스터는 DMMP 방식을 채용하여 선형 확장성

을 유지한다. 디스크를 공유함으로써 데이터베이스 엔진을 비롯한 소프트웨어 이식이 용이하도록 한다. 대규모 병렬처리 시스템보다 일반 상용 응용, 특히 OLTP 응용에 더 적합한 구조를 제공하며 다중프로세서 시스템보다 강력한 성능과 확장성을 제공한다.

Ⅲ. 클러스터 기반의 병렬처리 시스템

1. 시스템 구조

공유메모리 다중프로세서 시스템보다 강력한 처리능력을 제공하기 위하여 많은 프로세싱 노드들을 고유 상호연결망으로 연결하는 병렬처리 시스템들이 많이 사용되고 있다. 현재 대표적인 병렬처리 시스템들은 대부분 특정 응용 분야에 맞게 특화된 것이거나 매우 규칙적이고 시스템 전반에 걸쳐 균일성을 제공하는 연결망에 기반을 두고 있다[10] [11]. 이러한 구조는 매우 규칙적인 데이터 병렬성을 갖는 과학 연산 응용에 적합하며 이 분야에서 강력한 성능을 제공하고 있다. 그러나 이들은, 상요 응용들도 더욱 강력한 성능을 필요로 하고 있음에도 불구하고 규칙적 데이터 병렬성을 갖지 않는 특성 때문에 OLTP 분야들에서 널리 사용되지 못하고 있다.

기존 병렬처리 시스템보다 가격 대비 성능에서 우수한 잠재 특성을 갖고 있는 구조로서NOW(Network Of Workstation) 또는 POW(Pile Of Workstation) 구조가 최근에 연구되고 있다[13]. 이 구조는 아직 초기 단계로서 주류를 형성하기에는 몇 가지 문제점들을 안고 있다. 즉, 이 구조는 각 워크스테이션이 LAN으로 연결되어야 하므로 상대적으로 통신 지연시간이 크고 따라서 소프트웨어 최적화 정도에 대단히 민감하다. 이것은 대규모 연산 서버가 갖는 특성과 유사한 것이나 일반 상용 응용에서는 좋지 않은 특성이다. NOW 구조에서는 시스템 크기에 따라 통신 지연시간이 크게 증가한다. 이것은 한 LAN에 연결할 수 있는 적당한 시스템 수에 제한이 있는 것과 마찬가지로 시스템 확장성에 제한 요소가 된다. 실질적인 측면에서 운용에 관한 문제도 심각한 바, 사용자가 모두 다른 많은 워크스테이션들에 대한 동시 사용권 확보는 간단치 않은 문제이다.

따라서 OLTP 또는 유사한 데이터베이스 응용들에 대하여 우수한 성능을 제공하는 병렬처리 시스템으로서 새로운 구조가 필요하다. 이 구조는 공유메모리 다중프로세서가 제공해 온 메모리와 프로세서 자원의 효율성을 유지하고 기존의 많은 응용 프로그램들에 대하여 적절한 호환성을 유지하면서 공유메모리 다중프로세서 이상의 확장성을 제공할 수 있어야 한다. 또한, 시스템 크기가 커짐에 따라 고장 요인도 증가하나 상용응용에서는 과학 연산 응용에 비해 시스템 고장시 파급 효과가 매우 크므로 우수한 고장 감내 특성이 제공되어야 한다. 이러한 특성들은 다음과 같은 시스템 요구사항으로 정리할 수 있다.

- ○ 공유메모리 다중프로세서용 프로그램에 대한 호환성과 이식성
- ○ 기존 LAN보다 더 강력하고 효율적인 연결 구조
- ○ 병렬처리 시스템 수준의 확장성
- ○ 고장감내 특성

즉, 공유메모리 다중프로세서와 대규모 병렬처리 시스템의 혼합형 구조가 필요하며 두 특성을 서로 다른 계층에서 제공하는 계층 구조가 바람직하다. 본 논문에서는 이러한 계층 구조로서 확장성이 뛰어 난 클러스터 구조를 제안하고 있으며 이 구조는 전술한 NOW 구조에 비하여 더욱 효율적인 계층간 연결 구조를 제공하며 동시에 공유메모리 다중프로세서와 같은 이미지를 운용자와 사용자에게 제공한다.

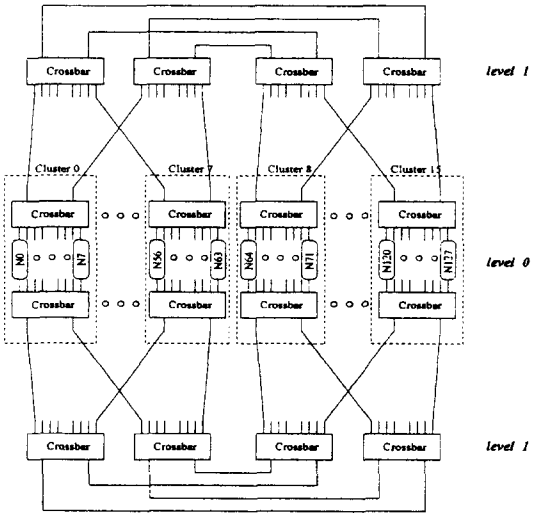


그림 2. SPAX 구조
Fig. 2. System Architecture of SPAX.

병렬처리 시스템이 갖는 확장성을 유지하면서 상용 응용에 적합하도록 제안된, 클러스터를 기반으로 하는

시스템(SPAX : Scalable Parallel Architecture based on Crossbar network) 구조는 그림 2와 같다. 그림 2에 나타난 바와 같이 SPAX는 16개의 클러스터를 이중 계층 크로스바 연결망(Xcent-Net)으로 연결한 구조를 갖는다. 이단 계층 구조를 제공함으로써 노드를 128개까지 연결할 수 있으며 이중 연결망은 시스템 신뢰도를 향상시키고 가용성을 높이고 있다. 클러스터는 8개 노드를 지원하고 상위 연결망(level 1)을 위하여 2개 통로를 제공한다. 클러스터 구성은 그림 3에 나타내었다. 한 클러스터는 프로세싱 노드(PN), 입출력 노드(ION) 및 통신접속 노드(CCN)들과 이들을 연결하기 위한 크로스바 연결망인 Xcent-Net으로 구성된다. 그림 3에 나타난 바와 같이 각 PN들은 공유메모리 다중프로세서 구조를 가지고 있으므로 공유메모리 다중프로세서 응용 프로그램들에 대하여 적절한 이식성과 호환성을 제공할 수 있다. 클러스터 내에서 디스크를 공유할 수 있도록 함으로써 공유메모리 다중프로세서를 대상으로 한 기존 상용 데이터베이스 응용들이 쉽게 이식될 수 있도록 하면서, 동시에 전 크로스바 연결망을 통하여 비 공유 디스크 방식 데이터베이스 응용도 효과적으로 지원할 수 있도록 하였다.

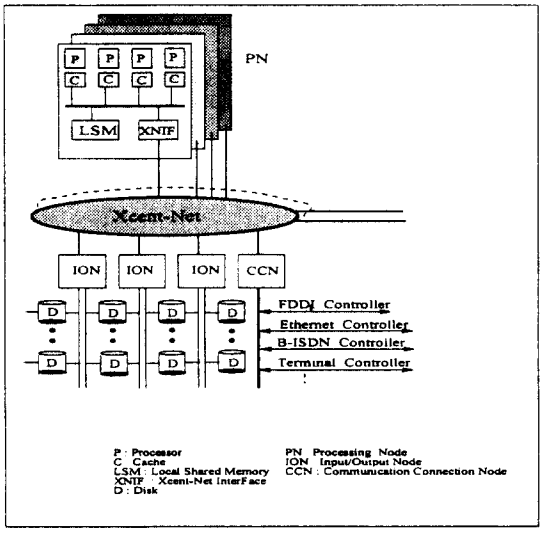


그림 3. 클러스터 구조
Fig. 3. Cluster Architecture of SPAX.

Xccent-Net은 클러스터당 2.67Gbytes/sec의 대역폭을 제공하여 LAN에 비하여 월등히 높은 성능을 나타내며 강력한 확장성과 효율성을 제공한다. 즉, 클러스터 내 연결망과 클러스터간 연결망이 동일한 구성소자와 프로토콜을 사용하여 확장성과 효율성이 높으며 소프트웨어가 단일 시스템 이미지를 유지하기 용이하다. 또 공유메모리 다중프로세서 노드를 연결하는 Xcent-Net은 공유메모리 다중프로세서 시스템들의 버스에 비하여 우수한 확장성을 통한 강력한 성능을 제공하면서도 적은 프로세서 시스템으로부터 점진적 확장이 가능하도록 한다. 이 특성은 본 논문에서 제안된 SPAX 시스템이 수퍼 컴퓨터에 비하여 매우 다양한 사용자와 응용 영역을 대상으로 한다는 점에서 매우 중요한 요건이다.

모든 노드는 인텔사의 Pentium Pro(P6) 프로세서를 하나 하나 이상 실장하고 있으며 Xcent-Net에 대한 접속 기능(XNIF)을 제공한다. XNIF는 PCI 버스를 통하여 노드의 다른 부분과 연결되며, 따라서 모든 노드는 PCI 접속 기능을 제공해야 한다. XNIF에 노드간 메시지 전송을 위한 버퍼와 제어 기능들이 구현되며, 이중 연결망 구조에 대한 사용 효율을 높일 수 있도록 고려되어 있다. Xcent-Net에서 정의된 패킷 전송 방식과 접속할 수 있도록 메시지를 패킷 단위로 분해, 조립하는 기능도 제공된다[14]

PN은 사용자 프로그램이 동작하는 기반으로서 각종 연산 처리 기능을 담당한다. PN은 Pentium Pro 프로세서를 4개 실장한 작은 규모의 공유메모리 다중프로세서를 형성한다. 지역 공유메모리(LSM)로서 1 Gbytes를 제공하며 캐시메모리는 Pentium Pro에 내장된 것으로 256 Kbytes 또는 512 Kbytes를 제공한다. 그 외에 지역 ROM과 시험을 위한 직렬 통신 기능들이 PCI 버스를 통해 지원된다.

ION은 디스크 또는 테이프 디바이스에 대한 입출력 기능을 제공하며 운영체제의 화일 서버가 동작하는 기반을 제공한다. ION은 Pentium Pro 프로세서를 하나 실장하나 필요에 따라 PN과 같이 4개까지 실장할 수 있다. 지역 메모리와 입출력을 위한 버퍼 메모리로서 256 Mbytes를 제공하며 최대1 Gbytes까지 확장할 수 있다. ION은 입출력 디바이스를 위하여 SCSI-II 접속 기능을 제공하며 PCI 버스를 통하여 연결된다. ION은 다양한 입출력 디바이스를 접속할 수 있어야 하므로 PCI 슬롯을 4개 이상 제공한다. CCN은 ION과 거의 같은 기능을 제공하며 PCI 버스를 통하여, SCSI-II 대신, 다양한 외부 통신 접속 기능을 제공한다.

그림 3에 나타난 구성은 권고 예이며 Xcent-Net은

각 노드 종류를 구분하지 않는다. 따라서 사용자 필요와 응용에 따라 구성은 변경될 수 있다. 즉, SPAX 시스템은 노드 단위로 확장이 가능하며 사용자 프로세서를 4개 갖는 최소 구성부터 그림 3과 같이 확장할 경우 256 사용자 프로세서까지 확장할 수 있다.[1] 한 클러스터의 최소 구성은 PN, ION, CCN 각 하나씩으로 구성될 수 있다.

2. 시스템 가용성

상용응용을 대상으로 하는 병렬처리 시스템에서 시스템 가용성은 매우 중요한 요구사항 중 하나이다. SPAX 시스템은 가용성을 높이기 위하여 다음과 같은 주요 특성들을 제공하며 단일점 고장(single point of failure)에 대처할 수 있도록 설계하였다[15].

- 프로세서쌍을 구성하여 프로세서 동작을 감시하는 기능(FRC : Functional Redundancy Checking)을 지원한다.
  이것은 소프트웨어 오버헤드 없이 효과적으로 프로세서 오 동작을 검출하기 위한 방법이다.
- 패킷 단위 오류 제어 방법을 제공한다.
  Xcent-Net을 통한 정보전달이 패킷 단위로 이루어지므로 한 클럭에 전송되는 정보에 대해 매 클럭 오류 검사를 수행하는 것만으로는 부족하다. 따라서 패킷 단위 오류 검출과 재 전송 방법을 제공한다.
- 대 용량 메모리에 대한 ECC 기능을 제공한다.
- 각 노드에 비휘발성 메모리를 제공한다.
  이 비휘발성 메모리는 시스템 형상 정보와 진단 기능들을 위한 것이다.
- 이중 연결망을 제공한다.
  Xcent-Net은 노드들과 달리 여분의 자원이 없으므로 단일점 고장에 대비하기 위하여 이중화 하고 있다.
- 전원 모듈은 주전원을 끄지 않고 교체할 수 있으며 여분의 전원 모듈을 제공한다.
- 각 노드는 전원 차단 없이 교체할 수 있다.
- 여분의 콘솔을 제공한다.
- 이중 디스크 접근 경로를 제공한다.

1) 그림 2, 그림 3과 같은 권고에 따른 클러스터 구성시 PN은 최대 64 노드가 된다. 따라서 사용자 프로그램을 수행하는 프로세서는 256개까지 실장될 수 있다. 그러나 이 역시 기준일뿐, 사용자 요구에 따라 유연하게 변경될 수 있다.

위에 열거된 특성들 중 이중 연결망은 대기(stand-by) 개념이 아니며 정상 동작 때는 두 연결망이 모두 사용되어 성능을 극대화 한다. 단일점 고장에 대비하기 위해서는 각 노드들도 둘이상 실장되어야 하며, 이 경우 최소 구성은 여섯 노드가 된다. 고장이 발생한 노드에 대한 조치(failover)는 소프트웨어에 의해 처리된다.

Ⅳ. 상호 연결망(Xcent-Net)

1. 구조

시스템 구조에서 살펴 본 바와 같이 SPAX 시스템 구조의 특징은 상호연결망에 의해 상당 부분 결정되고 있다. 따라서 시스템 설계 목표를 달성하기 위하여 최적의 상호연결망을 설계하는 것이 매우 중요하다. 연결망의 기본 구조를 결정하기 위하여 시스템 요구 조건과 기하학적인 특성들을 비롯한 여러 가지 변수들을 고려하여야 하며 다음과 같은 변수들이 주로 고려되었다.

- 소프트웨어에 대한 민감도
  연결망의 토폴로지에 따라 소프트웨어에 대한 민감도 차이가 매우 크다. 즉, CM-5와 같은 트리 연결망은 소프트웨어, 데이터와 프로세스 분배들에 따라 연결망 효율이 심하게 변하며 HiPi+Bus와 같은 공유 버스 구조는 영향을 거의 받지 않는다. SPAX 시스템은 과학 연산용 시스템과 같이 특정 알고리즘에 최적화 시킬 수 없으며 다양한 사용자와 부하 특성을 고려하여야 하므로 소프트웨어에 대한 민감도가 낮은 연결망을 설계하여야 한다.
- 대역폭과 전송 지연시간
  대역폭은 시스템의 병렬성을 충분히 지원할 수 있어야 한다. 단, 클러스터 기반 시스템이므로 클러스터 내와 클러스터간 연결을 위한 대역폭 배분은 비대칭이되 클러스터 내 대역폭을 우선 고려하여야 한다. 전송 지연시간은 사용자에 대한 응답시간에 관계되므로 대역폭보다 중요하다고 할 수 있다. 지연시간을 줄이기 위하여 연결망의 데이터 폭을 넓히고 동작 속도를 가능한 한 높이는 것이 좋으나 이것들은 구현 시 제약 조건에 따라 제한된다. 따라서 구조 측면에서는 프로토콜 변환이 없도록 하는 것이 중요하다. 연결망 종류에 따라 지연시간 특성도 달라지는 바, 여러 단을

거쳐야 되는 트리나 링 구조들에 비하여 한번에 연결이 가능한 버스나 크로스바가 유리하다고 할 수 있다. 그러나 버스는 전기 규격 특성에 따른 제약이 있어 대역폭에 한계가 있으며 트리 역시 스위치 단위 당 대역폭은 낮은 편이다.

○ 프로토콜 오버헤드

프로토콜이 매우 복잡할 경우 연결망 효율이 좋아진다고 하더라도 연결망 접속 기능이 복잡해지고 지연시간이 증가한다. 또한 복잡한 프로토콜은 구현과 검증이 어려우므로 구현 현실을 고려하여 프로토콜을 단순하게 하는 것이 바람직하다. 버스나 링 구조는 연결 매체를 공유하므로 프로토콜이 좀 더 복잡해 지며 점대점 구조로 갈수록, 또 경로 설정이 고정적일수록 단순한 프로토콜을 갖는다.

○ 구현 용이성

구현 환경은 설계된 연결망의 실현 가능성을 결정한다는 측면에서 이 변수는 매우 중요하다. 구현 측면에서는 VLSI로 구현할 수 있는 정도, 연결망 전체에 걸친 단순 반복성, 신호 특성을 유지할 수 있는 연결 방법들을 고려하여야 한다. 크로스바나 메쉬들은 반복형 구조로서 VLSI 구현에 적합하다. 그러나 하이퍼 큐브는 시스템 확장에 따라 제공해야 할 포트 수가 증가하므로 실장이 매우 어려워지고 링 구조는 특성상 매우 고속으로 신호를 전송하여야 하므로 인터페이스 소자 제작이 어렵다.

○ 상용 시스템들의 추세

상용 시스템들의 추세는 기술 추세를 반영한다고 보아 반드시 고려하여야 한다[16].

표 1에 각 연결망 구조와 위에서 기술한 변수들간 관계를 정리하였다.

표 1. 상호연결망 구조 비교
Table 1. Interconnection Network Comparison.

	Bus	Crossbar	Ring	MIN	Tree	Mesh	Hypercube
S/W 민간도	하	하	중	상	상	중	상
지연시간	저	저	중	중	고	고	중
구현 용이성	상	상	하	중	중	중	하
대역폭	하	상	중	상	하	중	중
프로토콜 복잡성	상	하	상	하	중	중	상

이와 같은 변수들을 고려할 때 Xcent-Net에 대한 주요 설계 항목들은 다음과 같다.

○ 클러스터 구조를 고려한 적절한 전송 프로토콜 및 중재 방법

○ 단일점 고장 허용 및 보드 고립화 기능

○ 구현 환경과 성능을 고려한 동작 속도, 포트 형태와 수 및 데이터 폭

○ 서로 떨어진 클러스터간 데이터 전송을 위한 동기 방식

위의 설계 항목들을 고려하여 설계된 Xcent-Net의 주요 사양은 다음과 같다.

○ 라우팅 및 중재 방법

Xcent-Net은Virtual Cut-through 라우팅을 사용하며 전송 단위인 패킷은 64 bytes 크기를 가진다. Wormhole 라우팅이 연결망 사용 효율면에서 유리할 수 있으나 클러스터 구조의 특성상 핸드쉐이크 신호들을 주고 받는 시간이 무시할 수 없는 정도이므로 신호 전송 속도를 상당히 낮추지 않으면 안된다. 따라서 Xcent-Net에서는 핸드쉐이크 오버헤드 영향을 최소화 하면서 성능을 높이기 위하여 Virtual Cut-through를 채택하였다. 각 패킷은 경로 제어 정보를 가지고 있고 각 Xcent 소자들이 경로 제어 기능을 내장하여 별도의 경로 제어용 신호선 없이 효과적인 분산 경로 제어와 중재를 수행한다.

○ 가용성

SPAX 시스템은 동일한 노드를 복수로 실장할 수 있으므로 한 노드의 고장에 대하여 가용성을 높일 수 있는 구조이다. 그러나 모든 노드들이 한 Xcent-Net은 고장이 발생할 경우 시스템 정상 운용될 수 없고, 따라서 시스템 가용성을 높이기 위하여 Xcent-Net에 단일점 고장 대책이 필요하다. Xcent-Net은 가용성을 높이기 위하여 이중 연결망 구조를 가지도록 설계되었으며 정상 동작 시에는 두 연결망을 모두 가동하여 두 배의 대역폭을 제공하고 있다. SPAX 클러스터는 다수의 동일한 보드들이 장착될 수 있으므로 보드 고장에 대비하고, 시스템 가용성을 높이기 위하여 Xcent-Net에서 보드 고립화 기능을 제공한다. 즉, 시스템 동작 중에 임의의 보드를 연결망에 연결하거나 제거할 수 있는 연결망 차원의 하드웨어 기능을 제공한다.

○ 대역폭

연결망에 대한 필요 대역폭은 부하 특성에 따라 크게 변할 수 있다. 그러나 구조 측면에서 보면 데이터 접근 경로가 프로세서 버스, PCI, Xcent-Net, 디스크로 구성되는 바, 물리적 한계를 갖는 디스크를 제외하면 PCI 보다 크거나 같은 대역폭을 제공하면 연결망 병목현상을 피할 수 있음을 알 수 있다. 현재 PCI는 133 Mbytes/sec의 대역폭을 제공하고 다수 사용자가 있는 버스 구조이므로 Xcent-Net은 포트 당 133 Mbytes/sec 또는 그 이상을 제공하면 충분하다. 실제는 이중 연결망 구조를 지원하므로 결과적으로 포트당 최대 266 Mbytes/sec, 최소 133 Mbytes/sec를 제공하고 있다. PCI는 두개 이상의 사용자(master)가 있음을 고려하면 이 대역폭은 PCI가 64bit로 확장될 때에도 여유가 있음을 알 수 있다.

○ 전송 동기 방식

일반적인 전송 동기 방식은 전역 클럭을 사용하는 것이다. 그러나 SPAX는 클러스터 구조이고, 각 클러스터간 거리가 수 미터 또는 그 이상이 될 수 있으므로 전역 클럭 구성 시 클럭 분배 및 동기 유지가 어렵다. 따라서 Xcent-Net은 SPAX 구조를 효과적으로 지원하고 upgrade 유연성을 제공하기 위하여 독립 동기 방식을 사용한다. 각 단위 연결망들이 독립 클럭을 사용하므로 데이터 전송 시 동기 신호가 필요하다. Xcent-Net은 근원지 동기 방식(source synchronous)을 지원하며, 송신측에서 데이터를 수신할 수 있는 동기 신호를 같이 전송한다. 데이터와 동기 신호가 같은 경로를 따라 전송되므로 스큐 문제와 경로 거리 제약을 피할 수 있고 upgrade에 유연성을 갖는다.

SPAX 시스템을 위하여 개발한 Xcent-Net은 10×10 크로스바 스위칭 소자(Xcent)를 기본으로 구성되는 이중 계층 크로스바 연결망으로서, 표 2와 같은 기능 및 성능 규격을 갖도록 설계되었다. 전술한 바와 같이 표 2와 같은 규격을 결정하는 데는 어느 정도 거리를 두고 배치될 클러스터 구조 및 구현 기술이 주요 제약 사항이었다. Xcent-Net은 구현 기술의 제약에 따라 바이트 슬라이스 구조를 가지며 클러스터 내 연결망 개략 구성은 그림4와 같다. 그림 4에서 XN0와 XN1은 동일한 연결망으로서 이중 연결망 구조를 지원하기 위한 쌍이다.

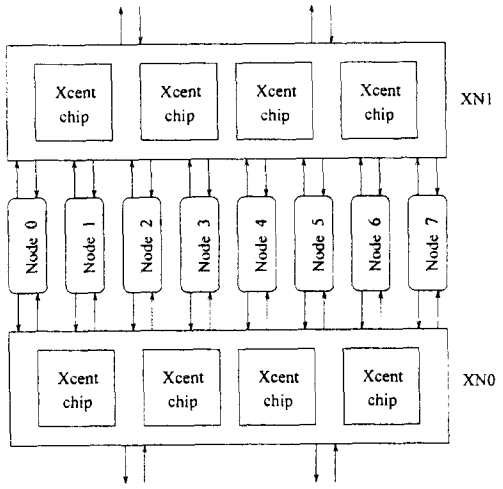


그림 4. 클러스터내 연결망 구성
Fig. 4. Block Diagram of Intra-cluster Network.

표 2. Xcent-Net 규격
Table 2. Xcent-Net Specification.

Xcent-Net	
토폴로지	이중화 2 계층 크로스바 연결망
최대 연결 노드	128 노드 (16 클러스터)
최대 연결 포트	입력 128 이중 포트, 출력 128 이중 포트
전송 데이터 폭	32-bit (4-byte) / 단일 포트
클럭 속도	33.3 MHz
최대 전송 대역폭	33.792 Gbytes/sec (4 bytes × 33 MHz × 256 ports)
결함 허용	단일점 결함 허용
입출력 전송 규칙	입출력 분리 전송
전송 단위	패킷 (packet)
경로 제어	Virtual Cut-through Routing
제어 알고리즘	적응 경로 제어
동기 방식	독립 동기
중재 방법	분산 경로 중재
망 제어 기능	보드 고립화 제어

2. 시뮬레이션

Xcent-Net을 중심으로 한 SPAX 클러스터 구조의 효율 및 성능을 이른 시기에 개략적으로 평가해 보기 위하여 SES/workbench[17]를 이용하여 사용율을 분석하였다. 본 시뮬레이션은 설계 초기 단계에서 주어진 프로세서와 노드 구조, 그리고 제어 메시지와 페이지 크기들에 대하여 Xcent-Net이 심각한 병목 현상을 일

으키는 가를 알아 보기 위한 것이다. 부하는 프로세싱 노드에서 발생시켰으며 Pentium Pro의 TPC-B 벤치마크 데이터를 기준으로 정하였다. 이때 TPC-B의 각 transaction 시작과 종료는 제어 메시지만을 사용하며 데이터 요구는 데이터 메시지를 사용하여 페이지 단위로 읽어 들인다고 가정하였다. Xcent-Net은 이중 연결망이 모두 정상 동작하는 경우를 대상으로 하였으며 그림5에 시뮬레이션 모델을 나타내었다. 그림 5의 가장 왼쪽에서 부하가 발생되고 가운데 부분이 Xcent-Net을 나타내는 크로스바 모델이다. 그 사이는 프로세싱 노드의 Xcent-Net 접속 부분을 모델링한 것으로 데이터 메시지 버퍼와 분리된 제어 메시지 버퍼를 갖고 있으며 이때 제어 메시지는 64 bytes, 데이터 메시지는 4 Kbytes 또는 8 Kbytes이다. 이들 메시지 크기는 현재 설계 중인 운영체제등 소프트웨어 사양을 반영한 것으로서 특히 데이터 메시지는 페이지 크기를 반영한 것이다. 프로세서 구조를 감안할 때 4 Kbytes보다 작은 페이지 크기를 운영체제가 지원하는 것은 생각하기 어려우므로 제외하였다. 오른쪽의 입출력 노드 모델은 실제 입출력 동작을 모델링한 것은 아니며 프로세싱 노드에서 발생된 메시지를 소모(consume)하는 기능을 갖도록 단순화한 노드 기능만 포함하고 있다.

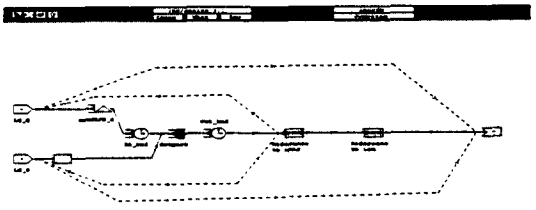


그림 5. Xent-Net 평가 모델
Fig. 5. Simulation Model of Xcent-Net.

그림 6은 데이터 메시지 요구율에 따른 Xcent-Net 사용율 변화를 나타낸 것이다. 제어 메시지 요구율을 변화시킨 결과도 있으나 영향이 미미하였다. 이것은 데이터 메시지가 제어 메시지에 비하여 64또는 128배나 크기 때문이며, 데이터 메시지는 그만큼 Xcent-Net 통로를 점유하는 시간이 많아 다른 메시지를 오래 대기시키기(blocking) 때문이다. 그림 6에서 데이터 메시지 크기를 두 배로 증가시킨 경우, 더욱 급격히 사용율이 증가함을 알 수 있다. 데이터 메시지 요구율과 크기가 Xcent-Net 사용 효율에 큰 영향을 미치고 있으며 데이터 메시지 크기가 되도록 작은 것이 전체 Xcent-Net 사용 효율면에서 유리함을 알 수 있다. 그러나 데이터 메시지가 운영체제의 페이지 크기에 비하여 현저히 작을 경우 페이지 로드나 스왑들에 대해 전송 지연 시간이 길어지므로 바람직하지 않다.

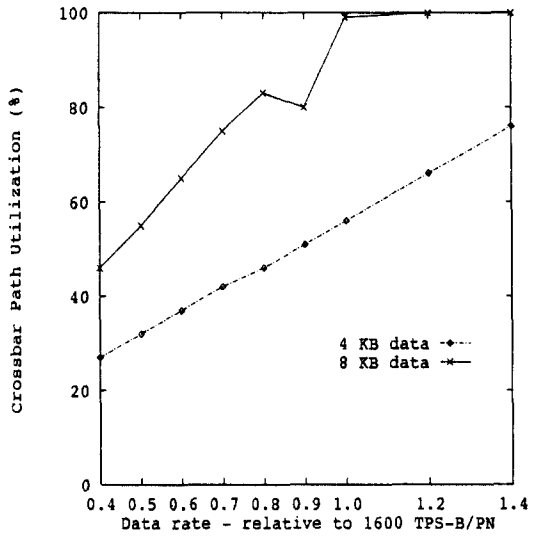


그림 6. Xent-Net 평가 - 메시지 비율에 따른 사용율
Fig. 6. Utilization of Xcent-Net.

그림 6에 따르면 현재 페이지 크기(4 Kbytes)에 대해서 각 프로세서가 최대 성능을 발휘하도록 부하가 걸렸을 때에도 Xcent-Net이 병목이 되지 않는 사양을 가지고 있음을 알 수 있다.

V. 결 론

본 논문에서는 상용 응용을 대상으로 하는 병렬처리 시스템으로서 이중 계층 크로스바 연결망을 갖는 클러스터 기반의 시스템을 제안하였다. 제안된 시스템 구조 설계를 위하여 시스템 구조 동향을 분석하였으며 상호연결망 구조들을 비교, 분석하였다. 제안된 시스템은 16개 클러스터를 연결했을 때 사용자 응용 프로그램을 위한 프로세서를 최대 256개까지 지원할 수 있었다. PCI 버스를 통하여 많은 입출력 디바이스와 다양한 외부 통신 접속 기능을 제공하고 있다. 높은 가용도를 제공하기 위하여 이중 연결망 구조를 채택하였으며 디스크 접근 통로도 이중화 하였다.

상호연결망 구조를 결정하기 위하여 목표 시스템의 응용 영역과 구현 환경들을 고려하였다. 설계된 상호연

결망 구조를 구현하기 위한 설계 항목들을 선정하고 성능 및 기능 규격들을 설계하였다. 설계된 연결망은 10×10 크로스바 소자를 기본으로 구성되며 클러스터당 2.67 Gbytes/sec의 대역폭을 제공한다. 클러스터간 거리가 수 미터 정도 될 것을 고려하여 신호 전송 지연 시간에 덜 민감한 Virtual Cut-through 방식 패킷 전송을 지원하며 각 클러스터 동작은 독립 동기 방식을 기반으로 한다.

제안된 클러스터 구조에서 설계된 상호연결망의 성능을 알아 보기 위하여 SES/workbench를 사용하여 시뮬레이션 하였다. 이 시뮬레이션을 통하여 설계된 계층 크로스바 연결망은 목표 시스템의 부하를 효과적으로 처리할 수 있음을 알 수 있었다. 크로스바 연결망은 충분한 병렬성을 가지고 있으므로 목표 시스템과 같은 응용에 매우 적합하나 한 통로의 대역폭은 구현 기술에 따른 제약이 있으므로 전송 데이터 크기에 영향을 많이 받음을 알 수 있었다. 즉, 전송 데이터 크기가 커지면 다른 메시지 전송을 방해하는 시간이 증가함으로 연결망 통로 확보를 기다리는 메시지 수가 증가하여 연결망 사용 효율이 급격히 떨어진다. 시뮬레이션에 따르면 4 Kbytes 크기를 갖는 데이터 메시지 전송에는 무리가 없음을 알 수 있었고 현 시스템 사양은 적절한 균형을 유지하고 있다고 판단되었다.

현재 연구는 설계를 마치고 구현 및 부분 시험 중이며 좀 더 다양한 경우에 대한 세밀한 특성 분석을 위한 시뮬레이션도 진행할 것이다. 앞으로 연구 방향으로는 신호 특성 분석 및 최적화들을 통하여 연결망 동작 속도를 높이는 것이 있다. 또한, 메시지 전송 시스템 평가를 위한 부하 특성에 관한 자료가 매우 부족한 바, 본 연구를 통하여 다양한 부하 특성 자료를 얻을 수 있을 것이다.

참 고 문 헌

[1] J.H. Edmondson, et al., "Superscalar Instruction Execution in the 2116A Alpha Microprocessor", *IEEE Micro*, May, 1995.

[2] G.Khermouch, "Technology 1994 : Large Computers", *IEEE Spectrum*, Vol. 31., No. 1., pp. 46-49, Jan., 1994.

[3] 김양우, 박진원, 임기욱, "병렬 DBMS 아키텍춰 분석", *정보과학회지*, 제13권, 제7호, pp 60-69, July, 1994

[4] E.E. Johnson, "Completing an MIMD Multiprocessor Taxatomy", *ACM SIGARCH Computer Architecture News*, Vol. 16, No. 3, pp. 44-47, June. 1988.

[5] Tom Manuel, "How sequent's new model outruns most mainframes", *Electronics*, pp. 76-79, May. 28. 1987.

[6] 윤용호, "다중처리 시스템: TICOM", *IEEE 한국 지부 병렬처리 컴퓨터 워크샵*, pp. 25-42, June, 1992

[7] 최성훈, 한우종, 윤석한, 안희일, "고속 중형 컴퓨터의 하드웨어 서브 시스템 개발", *대한전자공학회 전자계산 분과 학술 발표대회*, pp.218-220, May, 1993

[8] A.Gottlieb, et al., "The NYU Ultracomputer - Designing an MIMD Shared Memory Parallel Computer", *IEEE Trans. on Computers*, Vol. C-32, No. 2, pp. 175-189, Feb. 1983.

[9] D. Leonski, et al., "The DASH Prototype : Logic Overhead and Performance", *IEEE Trans. on Parallel and Distributed Systems*, Vol. 4, No. 1, pp. 41-61, Jan. 1993.

[10] Intel Co., *Paragon XP/S Product Overview*, 1991.

[11] A.Trew, G.Wilson, *Past, Present, Parallel*, Springer-Verlag, 1991.

[12] Gartner group, *Second Annual Advanced Technology Groups Conference*, 1992.

[13] T.E.Anderson, D.E.Culler, D.A.Patterson, "A Case for NOW", *IEEE Micro*, pp. 54-64, Feb. 1995.

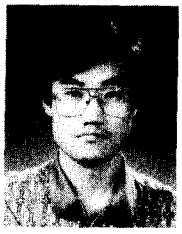
[14] 모상만, 신상석, 윤 석한, 임기욱, "고속병렬컴퓨터에서의 효율적인 메시지 전달을 위한 메시지 전송 기법", *대한전자공학회 하계학술 발표대회*, June, 1995

[15] 신상석, 민병호, 김정녀, 김중배, "SPAX의 고장 감내 기법", *한국정보과학회 컴퓨터시스템 연구회 학술 발표대회*, Sept., 1995

[16] 기안도, 한우종, 윤석한, "대규모 다중프로세서 실험시스템 조사(I)", *ETRI 주간기술동향*, 94-30

[17] Scientific Engineering Software, Inc., *SES workbench User's Manual*, Feb., 1992.

저 자 소 개



韓 宇 宗(正會員)

1959년 1월 2일생. 1981년 2월 고려대학교 전자공학과 학사. 1984년 9월 고려대학교 전자공학과 석사. 1995년 2월 고려대학교 전자공학과 박사. 1985년1월 ~ 현재 한국전자통신연구소 근무. 책임연구원. 주 관심분야 마이크로프로세서, 컴퓨터 구조, 병렬처리 구조, 상호연결망, 계층 메모리 구조등

林 基 郁(正會員)

1950년 8월 22일생. 1977년 인하대학교 전자공학과 학사. 1986년 한양대학교 전자계산학 석사. 1994년 인하대학교 전자계산학 박사. 1977년 ~ 1983년 한국전자기술연구소 선임연구원. 1983년 ~ 1988년 한국전자통신연구소 시스템소프트웨어 연구실장. 1988년 ~ 1989년 미 캘리포니아주립대학(Irvine) 방문연구원. 1989년 ~ 현재 한국전자통신연구소 시스템연구 부장. 책임연구원. 주 관심분야 소프트웨어 아키텍쳐, 컴퓨터 구조, 실시간 데이터베이스 시스템등

尹 碩 漢(正會員) 第 32卷 第 5號 參照

현재 한국전자통신연구소 근무