고성능 클러스터 시스템을 위한 인피니밴드 시스템 연결망의 설계 및 구현

모 상 만[†] · 박 경[††] · 김 성 남[†††] · 김 명 준[††††] · 임 기 욱[†††††]

요 약

인피니밴드(InfiniBand) 기술은 클러스터 컴퓨팅용 고성능 시스템 연결망으로의 활용을 목적으로 컴퓨터 업계를 중심으로 활발히 개발되고 있는 차세대 시스템 연결망 기술이다. 본 논문에서는 고성능 클러스터 시스템을 위한 인피니밴드 시스템 연결망의 설계와 구현을 다루며, 특히 이중(dual) ARM9 프로세서를 기반으로 한 인피니밴드 호스트 채널 어댑터(host channel adapter : HCA) 개발에 초점을 맞추어 기술한다. KinCA라는 코드명이 부여된 HCA는 클러스터 시스템의 각 호스트 노드(host node)를 하드웨어 및 소프트웨어적으로 인피니밴드 연결망에 연결한다. ARM9 프로세서 코어는 다중 처리기 구성을 위해 필요한 기능을 지원하지 않으므로, 두 개의 프로세서간 통신 및 인터럽트 메커니즘을 설계하여 KinCA 칩에 내장하였다. 일종의 SoC인 KinCA 칩은 0.18㎛ CMOS 기술을 사용하여 564핀 BGA(Ball Grid Array) 소자로 제작되었다. KinCA는 호스트 노드에 장착되어 송신과 수신 각각에 대하여 10Gbps의 고속 대역폭을 제공함으로써 고성능 클러스터 시스템의 구현을 가능하게 해준다.

Design and Implementation of an InfiniBand System Interconnect for High-Performance Cluster Systems

Sangman Moh[†] · Kyoung Park[††] · Sungnam Kim[†††]
Myung-Joon Kim[††††] · Kee-Wook Rim[†††††]

ABSTRACT

InfiniBand technology is being accepted as the future system interconnect to serve as the high-end enterprise fabric for cluster computing. This paper presents the design and implementation of the InfiniBand system interconnect, focusing on an InfiniBand host channel adapter (HCA) based on dual ARM9 processor cores. The HCA is an SoC called KinCA which connects a host node onto the InfiniBand network both in hardware and in software. Since the ARM9 processor core does not provide necessary features for multiprocessor configuration, novel inter-processor communication and interrupt mechanisms between the two processors were designed and embedded within the KinCA chip. KinCA was fabricated as a 564-pin enhanced BGA (Ball Grid Array) device using 0.18㎛ CMOS technology. Mounted on host nodes, it provides 10 Gbps outbound and inbound channels for transmit and receive, respectively, resulting in a high-performance cluster system.

키워드 : 클러스터 시스템(Cluster System), 시스템 연결망(System Interconnect), 인피니밴드(InfiniBand), 채널 어댑터(Channel Adapter), 멀티프로세서(Multiprocessor), SAN

1. 서 론

1946년 최초의 전자식 컴퓨터 ENIAC이 개발된 이후로 컴퓨터는 진화를 거듭하여 왔다. 인피니밴드(InfiniBand)는 컴퓨터 서버가 입출력 장치 및 타 서버와 상호 연결하는 방법을 획기적으로 개선한 새로운 시스템 연결망 기술이다. 인피니밴드 기술은 점대점 방식의 스위치 기반 연결망이다[1]. 인피니밴드 아키텍춰는 인피니밴드통상협회(InfiniBand Trade Association : IBTA)가 2000년 10월 발표한 규격서에 최초로 정의된 후 개정되어 오고 있다[2]. 인피니밴드의 어원은 '무한 대역폭'이란 단어로부터 유래되었다[1].

초기에 인피니밴드 기술은 고성능을 목표로 PCI(Peripheral Component Interconnect) 버스의 대체 기술로 출발하였다[3]. 그러나, 현재 인피니밴드 기술은 그 이상의 기술로서 서버 클러스터링에 필요한 여러 가지의 진보된 기능과 성능을 제공한다. 인피니밴드의 1차적인 목표는 PCI 버스의

† 종신회원 : 조선대학교 인터넷공학부 교수
†† 정 회 원 : 한국전자통신연구원 컴퓨터구조연구팀 팀장
††† 정 회 원 : 한국전자통신연구원 컴퓨터구조연구팀 선임연구원
†††† 종신회원 : 한국전자통신연구원 컴퓨터소프트웨어연구소 부장
††††† 종신회원 : 선문대학교 지식정보산업공학과 교수
논문접수 : 2003년 7월 10일, 심사완료 : 2003년 10월 21일

한계(성능 병목현상, 확장성 등)를 극복하는 것과 서버 클러스터링을 위한 시스템 연결 기술을 향상시키는 것이다[4-9].

인피니밴드 기술은 클러스터 컴퓨팅을 위한 통합 연결망을 목적으로 개발되고 있는 차세대 시스템 연결망 기술이다[10]. 인피니밴드의 범위는 소규모 서버 제품군에서부터 고성능 서버 솔루션에 이르기까지 다양하고 광범위한 시장(저장장치, 네트워킹, 서버 플랫폼, 메인프레인 컴퓨팅 등)을 포함한다[4, 9]. 다시 말해서, 인피니밴드 기술은 현존하는 기술 장벽의 극복은 물론 새롭고 향상된 기능 제공을 목적으로 설계된 정교한 시스템 연결망 기술이다.

본 논문에서는 고성능 클러스터 시스템을 위한 인피니밴드 시스템 연결망의 설계와 구현을 다루며, 특히 이중(dual) ARM9 프로세서를 기반으로 한 인피니밴드 호스트 채널 어댑터(host channel adapter : HCA) 개발에 초점을 맞추어 기술한다. KinCA라는 코드명이 부여된 HCA는 클러스터 시스템의 각 호스트 노드를 하드웨어 및 소프트웨어적으로 인피니밴드 연결망에 연결해준다. ARM9 프로세서 코어는 다중 처리기 구성을 위해 필요한 기능을 지원하지 않으므로, 두 개의 프로세서간 통신 및 인터럽트 메커니즘이 설계되어 KinCA 칩에 내장되었다. KinCA SoC 칩은 0.18㎛ CMOS 기술을 사용하여 564핀 BGA(Ball Grid Array) 소자로 제작되었다. KinCA는 호스트 노드에 장착되어 송신과 수신 각각에 대하여 10Gbps의 고속 전송속도를 제공함으로써 고성능 클러스터 시스템의 구현을 가능하게 해준다.

본 논문의 2장에서는 인피니밴드 기술의 개요와 아키텍춰를 간략히 기술하고, 3장에서는 계층적 구조, 다중 프로세서 이슈, 설계 검증과 DFT(Design For Test) 등을 포함한 KinCA의 설계에 관하여 상세히 기술한다. 4장에서는 KinCA의 구현 결과와 성능을 기술한다. 마지막 5장에서는 본 논문의 결론을 기술한다.

2. 관련 기술 및 연구

2.1 인피니밴드 기술의 출현

인피니밴드 아키텍춰는 관련 업계를 중심으로 표준화를 진행해 온 두 개의 연결망 기술인 NGIO(Next Generation I/O)와 FIO(Future I/O) 기술로부터 비롯되었다[4]. 1996년 인텔은 VI(Virtual Interface) 아키텍춰를 기반으로 한 입출력 기술 개발을 주도하였다. 이 기술은 NGIO라고 명명되었고, 델 컴퓨터, 히다찌, 인텔, NEC, 후지쯔, 지멘스, 선 등이 기술 개발에 참여하였다. 거의 동일한 시점에, 또 하나의 표준화 노력으로서 FIO가 IBM, 컴팩, 어댑텍, 3Com, 시스코, HP 등을 중심으로 개발되기 시작하였다. NGIO는 대량의 소규모 서버에 초점을 맞추었고, 반면에 FIO는 고성능 플랫폼에 초점을 맞추었다[4, 9]. 이후 양 진영은 업계의 양분이 바람직하지 않다는 것을 인식하고, 1999년 여름 양 진영을 통합하는 계획을 수립했고, 뒤이어 1999년 10월 IBTA를 결성하였다.

2000년 10월 IBTA는 개발자 회의를 개최하고 인피니밴드 표준 규격서(버전 1.0)를 완성하여 업계에 공표하였다[2]. 인피니밴드 아키텍춰(InfiniBand architecture : IBA)는 서버 내부에서의 프로세서간 클러스터링 연결은 물론이거니와 서버를 원격 저장장치나 네트워킹 장치 등에 연결할 수 있게 해준다[1]. 인피니밴드 표준 규격서는 현재도 지속적으로 보완 및 개정되고 있다[2].

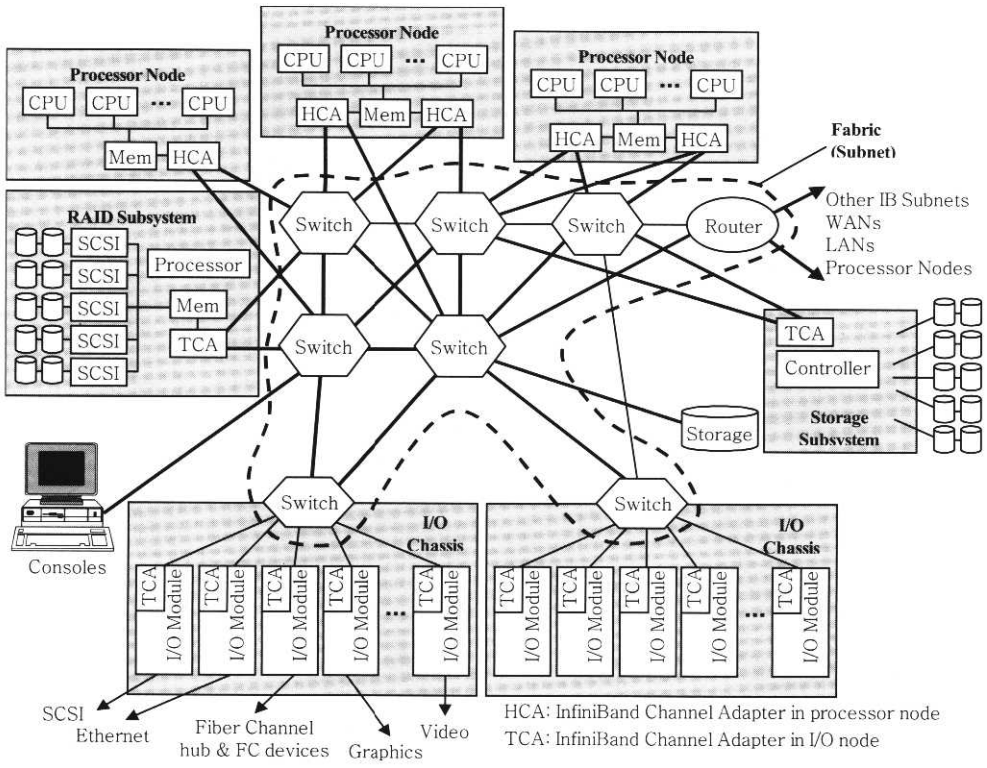
버스와 달리 IBA는 완전 이중 직렬 링크(full duplex serial link)를 사용함으로써 동시에 양방향 전송을 가능하게 하여 유효 처리율을 두 배로 증가시킨다. 보다 높은 대역폭을 제공하기 위하여 복수 개의 직렬 링크를 병렬로 연결하는 것 또한 가능하다. 이를 위해 IBA는 1x, 4x, 12x 포트를 정의하여 2.5 Gbps 직렬 링크의 전송 대역폭을 각각 1배, 4배, 12배 향상시킬 수 있다. IBA는 스위치 기반 연결망이므로 포트 속도는 노드 수와 무관하여 연결되는 장치의 수를 제한하지 않는다[4].

2.2 인피니밴드 아키텍춰

인피니밴드 연결망에는 4개의 기본 구성요소가 있다. 이들은 호스트 채널 어댑터(HCA), 타겟 채널 어댑터(target channel adapter : TCA), 스위치(switch), 그리고 라우터(router)이다. HCA는 서버 내에 존재하는 인터페이스로서, IBA 연결망은 물론 서버 메모리 및 프로세서와 직접 통신한다. TCA는 스토리지 어댑터와 네트워크 어댑터 두 종류로 크게 분류되며, 호스트 컴퓨터와 입출력 장치 사이의 일종의 브리지 역할을 수행한다. 스위치는 HCA와 TCA를 연결하여 서브넷을 구성하고 네트워크 트래픽을 제어한다. 라우터는 서브넷과 타 서브넷을 연결해 주는 일종의 브리지이다[1, 4-5]. (그림 1)은 인피니밴드 기반 시스템 연결망의 구성 예를 나타낸 것이다.

인피니밴드 연결망 내부에 연결된 각 종단 노드들은 서로 통신할 수 있다[7-8]. 응용 프로그램은 메시지 송신 또는 수신을 요구하고, 일종의 인터페이스 규약인 '버브(Verb)'를 통하여 송수신 완료 정보를 얻는다. HCA나 TCA는 작업 큐 항목(work queue element)을 검색하여 해당 메시지를 호스트 메모리로부터 읽어온 후, 패킷화하여 연결망으로 송신한다. 수신 패킷은 메시지로 조합되어 작업 큐에 저장된다. 노드는 해당 완료 큐 항목(completion queue element)을 검색함으로써 요청한 메시지의 완료 상태를 알 수 있다.

IBA에는 4개의 계층인 물리 계층, 링크 계층, 네트워크 계층, 트랜스포트 계층이 존재한다[4-9]. 인피니밴드 물리 계층은 케이블, 커넥터, 동작 중 교체(hot swap) 기능을 포함한 IBA의 전기 및 기계적인 특성을 정의하고 있다. 링크 계층



(그림 1) 인피니밴드 시스템 연결망

은 IBA의 중추로서, 패킷 구성, 점대점 링크 명령어, 서브넷 내에서의 스위칭, 데이터 건전성 등을 포함하고 있다. 네트워크 계층은 하나의 서브넷에서 타 서브넷으로의 라우팅을 담당한다. 트랜스포트 계층은 다양한 종류의 메시지 및 패킷 전달 순서를 처리할 뿐만 아니라 분할, 다중화, 신뢰성 연결 및 데이터그램을 지원하는 트랜스포트 서비스를 수행한다[1, 4-5].

고성능 뿐만 아니라 고신뢰도, 고가용성, 우수한 서비스 능력 또한 고성능 서버의 주요 요구사항에 속한다[11]. 100개를 상회하는 PCI 버스 신호선에 비하여, 2.5 Gbps 인피니밴드 링크는 4개의 신호선으로 구성되어 있다. 또한, IBA는 각 장치에 다중 포트를 지원하여 하나의 수신처에 대하여 다중의 경로를 제공한다. 아울러 진보된 오류 검출 능력을 제공하기 위하여 다중 CRC를 사용한다. IBA 연결망은 데이터 전달의 가용성을 높이기 위하여 다중 경로를 제공하는 고유의 중복성을 갖고 있으며, 자동 경로 이주(automatic path migration) 메커니즘이 제공된다. IBA에서의 서비스 능력을 향상시키기 위하여, 입출력 장치에 대하여 동작 중 교체 기능을 제공함으로써, 동작 중에 하나의 입출력 장치를 탈착한 후 새로운 입출력 장치를 장착할 수 있게 해준다[1, 4].

3. KinCA의 설계

인피니밴드 HCA인 KinCA는 호스트 노드와 인피니밴드 연결망 사이에서 패킷을 생성하고 소비하는 일종의 IBA 장치로서, 호스트 인터페이스, 트랜스포트 계층, 네트워크 계층, 링크 계층, 물리 계층 블록들로 구성되어 있다. 본 절에서는 KinCA의 설계와 검증 이슈를 기술하며, 특히 계층 구조, 다중 프로세서 이슈, 설계 검증에 초점을 맞춘다.

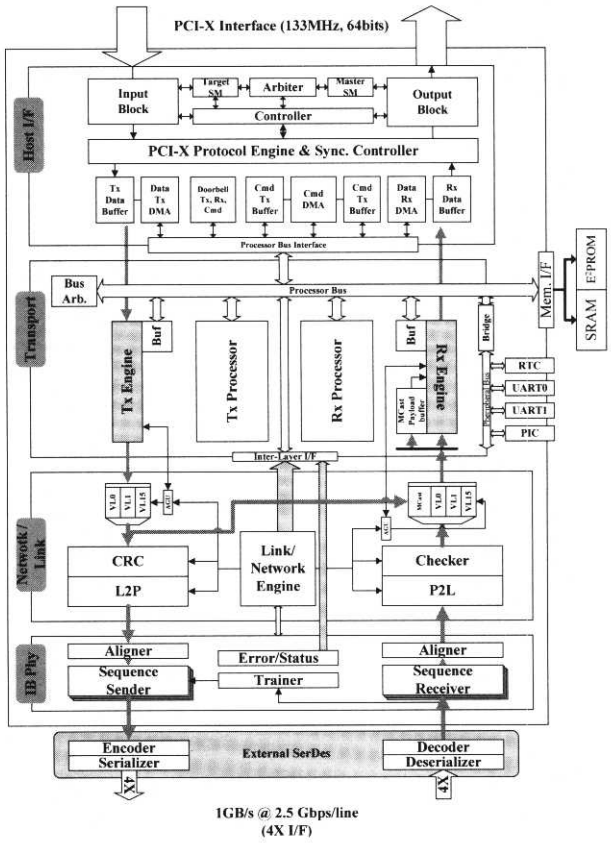
3.1 계층적 구조

KinCA는 (그림 2)에 나타낸 바와 같이 호스트 인터페이스, 트랜스포트 계층, 네트워크/링크 계층, 물리 계층 등의 계층적 구조로 이루어져 있다. KinCA는 메시지 송신과 수신을 동시에 수행할 수 있으며, 송신 흐름과 수신 흐름은 전체 프로토콜 스택을 통하여 서로 분리되어 있다.

호스트 인터페이스는 명령어와 메시지를 처리하는 프로토콜 엔진 뿐만 아니라 호스트 메모리로부터 메시지를 읽어오거나 저장하는 DMA 엔진을 포함한다. 본 설계에서의 호스트 버스는 업계 표준인 PCI-X 버스이며, 호스트 인터페이스는 PCI-X 버스와 인피니밴드 트랜스포트 계층 사이의 브리지로 간주될 수 있다.

트랜스포트 계층은 인피니밴드 트랜스포트 프로토콜을 하드웨어 및 소프트웨어적으로 처리한다. 소프트웨어 처리를 위하여 두 개의 ARM9 프로세서가 KinCA 칩에 내장되어 있으며, 각각 송신 프로토콜과 수신 프로토콜을 처리한다. 송신 프로세서와 송신 엔진은 도어벨(doorbell)을 통하여 호스트 인터페이스로부터 메시지 송신 요구를 받고, DMA

인터페이스를 이용하여 호스트 메모리로부터 데이터를 읽어와 인피니밴드 트랜스포트 프로토콜에 따라 패킷화(packetizing)한 후, 패킷을 네트워크/링크 계층으로 내려보낸다. 한편, 수신 프로세서와 수신 엔진은 네트워크/링크 계층으로부터 올라온 패킷을 해체(depacketizing)하여 검증하고, DMA 인터페이스를 이용하여 수신한 데이터를 호스트 메모리에 저장한 후, 최종적으로 도어벨을 통하여 메시지 수신 사실을 호스트 인터페이스에게 알려준다. 여기서, 메시지는 하나 이상의 패킷으로 구성되어 있다. KinCA 칩 내부에 있는 두 개의 송수신 프로세서는 초기 설정 및 메시지 전송 동작 중에 상호 통신하고 협력해야 함에도 불구하고, ARM9 프로세서는 기본적으로 다중 처리를 위한 기능을 제공하지 않는다. 따라서, 프로세서간 통신 및 인터럽트 메커니즘의 설계가 필요하며, 이에 대한 내용은 3.2절에서 기술하기로 한다.



(그림 2) KinCA의 아키텍춰 구성

네트워크/링크 계층은 인피니밴드 프로토콜의 네트워크 및 링크 계층을 처리하며, 데이터 패킷은 물론 연결망 관리 목적의 링크 패킷을 제어한다. 교착상태를 막기 위하여 버퍼링, 흐름 제어, 그리고 3개의 가상 레인(데이터용 2개와 관리용 1개)이 송수신 각 방향에 제공된다. 또한, 하드웨어 수준에서의 오류 제어를 위하여 CRC 생성기와 검사기가 포함되어 있다. 네트워크/링크 계층을 FSM(Finite State Machine) 방식으로 설계함으로써, 매우 빠른 프로토콜 처리를 가능하게 하였다.

〈표 1〉 KinCA의 기능 및 성능 규격

호스트 인터페이스	
PCI-X	64bits, 133MHz
Host DMA	3 engines(Tx/Rx/command)
Host doorbell	12 doorbells
트랜스포트 계층	
Transport service	RC, UC, RD, UD, Raw Packet
Transport engine	Separate transmit/receive
Protocol engine	2 ARM922T processors
네트워크/링크 계층	
Network/link engine	Separate transmit/receive
Virtual lanes	Two for data, one for management
Buffer size per virtual lane	16KB for data, 4KB for management
MTU	Maximum 2KB
물리 계층	
Transmission mechanism	8b/10b serial transmission
Transmission speed	4×2.5Gbps(10Gbps)

물리 계층은 비트 열(bit stream)을 인피니밴드 연결망으로 송수신하는 기능을 수행한다. 링크 트레이너(link trainer), 8b/10b 변환기, 비트 정렬기(bit aligner), 패킷 순서 송수신기, 그리고 오류 및 상태 제어기 등이 직렬/병렬 변환기(serializer/deserializer)와 더불어 물리 계층의 역할을 수행한다. 송신 채널과 수신 채널은 각각 인피니밴드 패킷을 최대 10 Gbps의 속도로 전송한다.

<표 1>은 호스트 인터페이스에서부터 물리 계층에 이르기까지 KinCA의 기능 및 성능 규격을 나타낸 것이다.

비교적 복잡한 트랜스 프로토콜은 다양한 호스트 응용에 따라 메시지의 스케줄링과 제어를 지원하기에 충분히 유연해야 하므로, 트랜스포트 프로토콜의 많은 부분이 이중 ARM9 프로세서 코어[12]를 이용하여 처리된다. 트랜스포트 계층을 제외한 다른 모든 계층은 효율성과 성능을 고려하여 하드웨어로 설계되었다. 모든 계층을 통하여 송신 흐름과 수신 흐름은 최대한 서로 분리되어 있다. 그 이유는 송신 동작과 수신 동작은 각 방향에서 10Gbps 네트워크 성능을 달성하기 위하여 동시에 수행되어야 하기 때문이다. DMA 엔진을 포함한 호스트 인터페이스의 많은 부분 역시 송신 부분과 수신 부분이 서로 분리되어 있다.

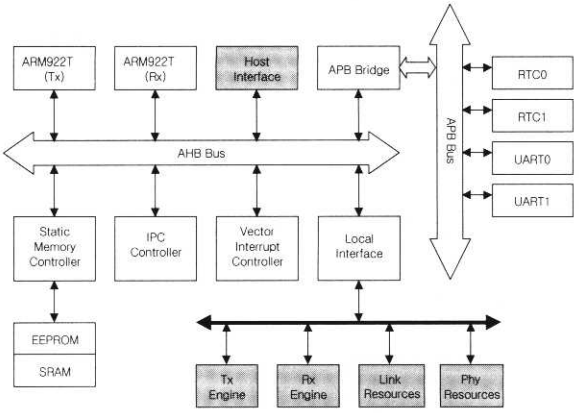
3.2 다중 프로세서 이슈

MTU(Maximum Transfer Unit) 크기의 2Kbytes 패킷을 8b/10b 코딩 방식의 10Gbps 채널을 통하여 전송하는 경우에, KinCA는 대략 1초에 최대 50만 개의 패킷을 처리해야

한다(10Gbps / (2Kbytes/packet×8bits/byte×10b/8b)≈500K packets/sec). 더욱이 KinCA는 송신 흐름과 수신 흐름을 동시에 처리하기 때문에, 50만 개의 송신 패킷과 50만 개의 수신 패킷을 동시에 처리해야 한다. 목표 시스템에서 하나의 ARM9 코어는 200MIPS의 성능을 제공하므로, 각 패킷에 약 200IPS(Instructions Per Second)가 할당될 수 있다. 이는 트랜스포트 프로토콜 처리에 다소 부족하며, 더욱이 구현 제약에 의해 실질적인 패킷당 성능은 이보다도 감소된다.

따라서, 단일 프로세서 대신에 이중 ARM9 프로세서 코어를 송신 처리와 수신 처리 각각에 별도로 사용한다. 이렇게 함으로써 패킷당 성능이 약 두 배로 증가한다. 이와 같은 이중 프로세서 코어 기반 설계는 두 가지의 장점을 가져온다. 그 중 하나는 송신 흐름과 수신 흐름의 분리 수행에 따른 소프트웨어 설계의 간소화이고, 다른 하나는 별도의 실시간 운영체제가 요구되지 않는다는 것이다. 단일 프로세서 코어를 사용하여 독립적인 송신 흐름 제어 작업과 수신 흐름 제어 작업을 통합 수행하기 위해서는, 두 작업 사이의 프로세스를 관리하기 휘한 소규모 실시간 운영체제가 요구된다. 그러한 운영체제는 소프트웨어 개발을 용이하게 하지만, 시스템 성능을 저하시키는 오버헤드를 유발한다. 송신 및 수신 프로토콜 처리 루틴을 분리된 두 개의 프로세서 코어에서 각각 수행함으로써, 운영체제가 더 이상 요구되지 않으며 부가적인 성능 감소가 제거된다.

내장된 이중 ARM9 프로세서 코어[12]는 내부 버스를 통하여 메모리와 주변 장치를 공유하며 결과적으로 다중 프로세서 시스템을 구성한다. (그림 3)에 보여지는 것처럼, 두 개의 ARM9 프로세서 코어가 AHB 버스[15]를 통하여 메모리를 포함한 고성능 주변 장치[13-14]에 연결된다. 여기서, 32 비트 AHB 버스는 주소 버스와 데이터 버스가 분리되어 있으며 다중 버스 마스터(bus master)를 허용하기 위하여 버스 중재 기능을 갖는다. 한편, 낮은 성능의 주변 장치[16, 17]는 APB 버스에 연결된다. AHB 버스와 APB 버스는 APB 브리지를 통하여 소프트웨어에 투명하게 상호 연결되어 있다.



(그림 3) 공유 버스 기반 이중 프로세서 시스템

내부 버스인 AHB 버스와 APB 버스에 연결되는 장치들은 두 개의 ARM922T 프로세서 코어, 호스트 인터페이스 블록, 벡터 인터럽트 제어기, SRAM 제어기, 프로세서간 통신(IPC) 제어기, 송신 및 수신 엔진과의 로컬 인터페이스, APB 브리지, 두 개의 실시간 클럭(RTC), 그리고 두 개의 UART 등이다.

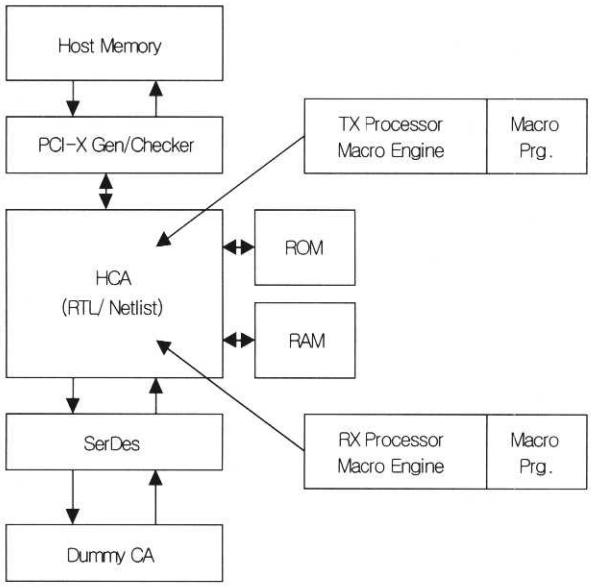
메모리를 포함한 모든 주변 장치들은 두 개의 ARM9 프로세서에 의하여 공유된다. 이와 같은 구성은 일종의 대칭형 다중 프로세서이다. 다중 프로세서 시스템에서는 특정 프로세서 식별자, 프로세서간 통신, 프로세서간 인터럽트 등이 필수적으로 요구된다. 그러나, ARM9 프로세서는 그와 같은 기능을 전혀 지원하지 않는다. 따라서, 본 연구에서는 다음과 같은 기능을 새로이 설계하였다.

- 프로세서 식별자 할당 : 시스템 초기화 후에 특정 프로세서 식별자를 부팅 순서에 따라 각 프로세서에 할당한다.
- 프로세서간 통신 : 프로세서간 통신을 지원하기 위하여 통신 요구 명령어와 메시지를 포함하는 메일 박스 메커니즘을 제공한다.
- 프로세서간 비동기 인터럽트 : 프로세서간 비동기 인터럽트를 처리하기 위하여 프로그램 가능한 비동기 인터럽트 하드웨어를 제공한다.
- 인터럽트 재지정 : 벡터 인터럽트 처리에 있어서 프로세서간 부하 균형을 위하여 인터럽트를 처리할 프로세서를 재지정하는 메커니즘을 제공한다.

3.3 설계 검증

KinCA 칩은 400만 개 이상의 게이트를 집적한 고밀도 SoC로서, 설계 검증이 매우 복잡하다. 검증 시간을 줄이고 컴퓨팅 자원을 최소화하기 위하여, ARM9 프로세서를 모사한 마크로 엔진을 개발하였으며, 이것은 KinCA 설계의 기능 검증을 위하여 특별히 고안된 것으로서, ARM9 프로세서의 간략화된 버전이다. 메모리 모델을 포함한 다른 시뮬레이션 모델들도 개발되었다. (그림 4)는 KinCA 설계 검증에서 사용된 기능 검증 모델들을 나타낸 것이다.

DFT 기능 제공을 위하여 ARM사에서 제공한 ARM9과 주변 장치 로직, 메모리 BIST(Built-In Self Test), ATPG (Automatic Test Pattern Generation)와 연계된 내부 완전 주사(internal full scan), JTAG 경계 주사(boundary scan) 등이 KinCA 설계에 포함되었다. ARM9 프로세서와 내부 버스에 연결된 주변 장치의 검증 과정에서는 ARM사에서 제공한 TIC 벡터[15]를 사용하였다. 설계 검증 환경은 네트리스트 검증과 타이밍 분석에 부분적으로 사용되었고 KinCA 칩의 시험 패턴으로도 활용되었다.

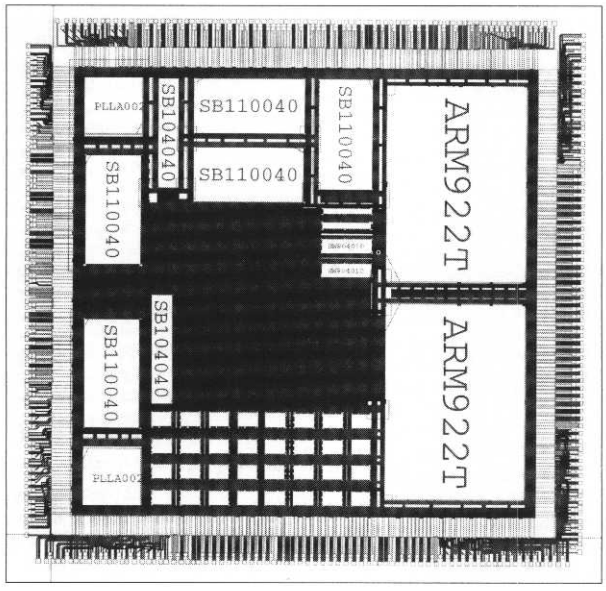


(그림 4) 기능 검증 모델

4. 구현 결과

설계 및 검증을 마친 후, KinCA 칩은 0.18㎛ 4계층 금속 CMOS 표준 셀 기술을 이용하여 제작되었다. (그림 5)는 10mm×10mm 다이에 구현된 KinCA 칩의 배치 설계 결과를 보여준다. KinCA 칩의 구현 규격은 <표 2>에 상세하게 나타나 있다.

KinCA 칩은 (그림 6)에 보여지는 것처럼 564핀 GBA(Ball Grid Array) 소자로 제작되었다. KinCA 칩은 (그림 7)에 나타낸 바와 같이 PCI-X 슬롯을 가진 PCI 카드(보드) 상에 장착된다. 여러 장의 KinCA 카드를 사용함으로써, 다중

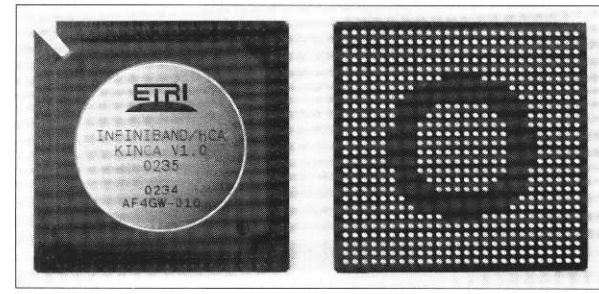


(그림 5) KinCA 칩의 배치 설계 결과

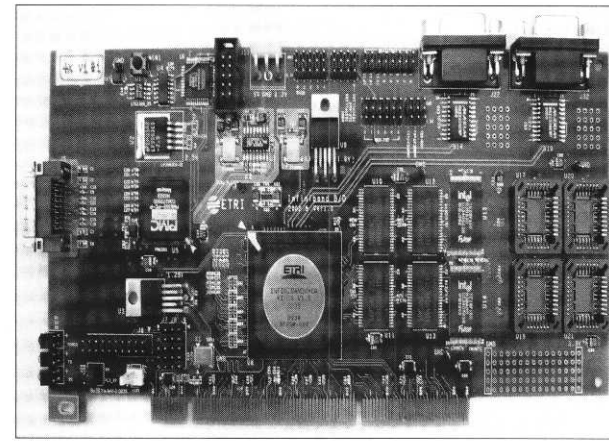
〈표 2〉 KinCA 칩의 구현 규격

Technology	0.18㎛ 4-layer metal CMOS standard cell	
Package	564 enhanced BGA	
Die	10mm×10mm	
Gates	ARM922T(8.1mm^2)	430K×2 = 860K
	Memory	1,663K
	PLL	113K×2 = 223K
	logic	1,611K
	Total	4.358 Million
Power supply	ARM9 core, logic	1.8Volt
	IO(LVTTL)	3.3Volt
	IO(SSTL)	2.5Volt
Clock	PCI-X logic	133MHz
	UART	7.3728MHz
	RTC	1MHz
	ARM9 core	125MHz
	SerDes interface	250MHz
Power dissipation	3.5 Watt in total	

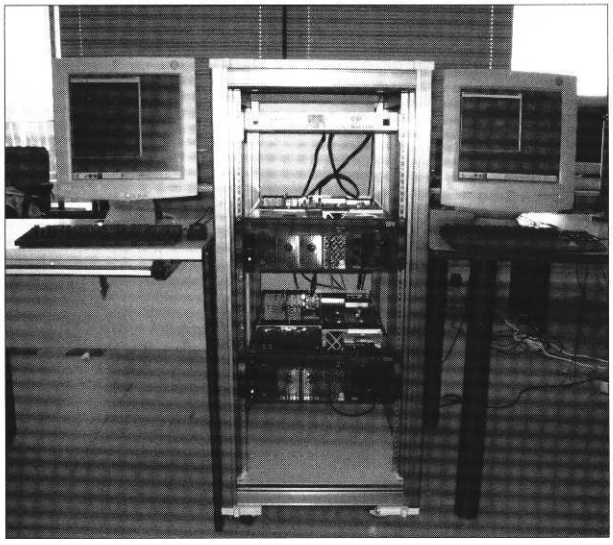
노드의 고성능 클러스터 시스템을 구성하기 위하여 여러 개의 호스트를 인피니밴드 연결망을 통하여 상호 연결할수 있게 된다. (그림 8)은 KinCA 카드를 장착한 두 개의 노드가 인피니밴드 연결망을 통하여 상호 연결되어 있는 KinCA 기반 클러스터 시스템의 테스트베드를 보여준다.



(그림 6) KinCA 칩



(그림 7) KinCA 카드



(그림 8) KinCA 기반 클러스터 시스템의 테스트베드

5. 결 론

본 논문에서는 이중 프로세서 코어를 기반으로한 인피니밴드 HCA인 KinCA의 설계와 구현을 기술하고 논의하였다. ARM9 프로세서 코어가 다중 프로세서 구성을 위하여 필요한 기능을 제공하지 않기 때문에, 프로서간 통신과 인터럽트 메커니즘이 새롭게 설계되어 칩에 내장되었다. 많은 DFT 기능과 연계하여 깊이 있게 수행한 설계 검증은 KinCA의 기능과 성능을 크게 향상시켰다. KinCA 칩은 0.18㎛ 4계층 금속 CMOS 기술로 구현되어 564핀 BGA 소자로 제작되었다. 호스트 노드에 장착된 KinCA는 송신과 수신 각각에 대하여 10Gbps 전송 속도를 제공함으로써, 고성능 클러스터 시스템 구성을 가능하게 해준다.

참 고 문 헌

[1] JNI Corporation, "An Introduction to InfiniBand : Bridging I/O up to Speed," *White Paper*, http://www.jni.com/Products/ib.cfm, Nov., 2001.

[2] InfiniBand Trade Association (IBTA) Official Homepage, http://www.infinibandta.org/, Mar., 2003.

[3] D. Pendery and J. Eunice, "InfiniBand Architecture : Bridge over Troubled Waters", *Research Note*, http://www.infinibandta.org/newsroom/whitepapers/, April, 2000.

[4] W. T. Futral, *InfiniBand Architecture Development and Deployment : A Strategic Guide to Server I/O Solution*, Intel Press, 2001.

[5] T. Shanley, *InfiniBand Network Architecture*, Edited by J. Winkles, Addison-Wesley, 2002.

[6] 모상만, "인피니밴드 기술 현황과 전망", 전자신문 테마특강, 2001.

[7] 박 경, 모상만, "InfiniBand : 차세대 시스템 연결망", 정보과학회지, 제19권 제3호, pp.43-51, 2001.

[8] 모상만, 김용연, "고성능 컴퓨터 기술진화 및 차세대 클러스터 연결망 개발동향", 주간기술동향, 통권 1000호, pp.93-107, 2001.

[9] S. Moh, Y. Y. Kim, S.-H. Yoon, M.-J. Kim and K.-W. Rim, "InfiniBand Technology : The Next Generation System Interconnect," *Proc. of the 1st World Korean Business Convention*, pp.C.2-8, Oct., 2002.

[10] A. Hartmann, "InfiniBand Architecture : Evolution/Revolution InfiniBand Adoption in the Enterprise Datacenter," *White Paper*, http://www.vieo.com/vieo_whit.html, July, 2001.

[11] Russell and J. Steczkowski, "Introduction to the InfiniBand Architecture," *White Paper*, http://www.crossroads.com/govt/whitepapers.asp, April, 2000.

[12] *ARM922T Technical Reference Manual*, ARM Ltd., 2000.

[13] *ARM PrimeCell Vector Interrupt Controller (PL190) Techincal Reference Manual*, ARM Ltd., 2000.

[14] *ARM PrimeCell Static Memory Controller (PL092) Techincal Reference Manual*, ARM Ltd., 2000.

[15] *AHB Example AMBA System Techinca Reference Manual*, ARM Ltd., 1999.

[16] *ARM PrimeCell RTC (PL031) Techincal Reference Manual*, ARM Ltd., 2000.

[17] ARM PrimeCell UART (PL011) Techincal Reference Manual, ARM Ltd., 2000.

모 상 만

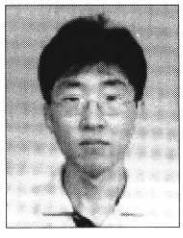
e-mail : smmoh@chosun.ac.kr

2002년 한국정보통신대학원대학교(ICU) 공학부(박사)

1991년~2002년 한국전자통신연구원(ETRI) 컴퓨터소프트웨어연구소(팀장)

2002년~현재 조선대학교 인터넷공학부 교수

관심분야 : 컴퓨터시스템, 병렬컴퓨팅, 모바일컴퓨팅



박 경

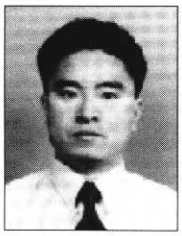
e-mail : kyoung@etri.re.kr

1991년 전북대학교 컴퓨터공학 학사

1993년 전북대학교 컴퓨터공학 석사

1993년~현재 한국전자통신연구원 컴퓨터구조연구팀 팀장

관심분야 : 컴퓨터구조, 마이크로프로세서 구조, 상호연결망, SoC



김 성 남

e-mail : ksn@etri.re.kr

1991년 고려대학교 전자전산학과(학사)

1993년 고려대학교 전자공학과 반도체공학 전공(공학석사)

1998년 고려대학교 전자공학과 반도체공학 전공(공학박사)

1999년~현재 한국전자통신연구원 컴퓨터구조연구팀 선임연구원

관심분야 : SoC ASIC 설계, 클러스터 및 컴퓨터 구조, 고성능 연결망 기술

김 명 준

e-mail : joonkim@etri.re.kr

1978년 서울대학교 자연과학대학 계산통계학과(이학사)

1980년 한국과학기술원 전산학과(이학석사)

1986년 프랑스 Nancy 제1대학교 응용수학 및 전산학과(이학박사)

1980년~1981년 아주대학교 종합연구소 연구원

1981년~1986년 프랑스 Nancy 전산학연구소(CRIN) 연구원

1993년 프랑스 Univ. of Nice Sophia-Antipolis 방문교수

1986년~현재 한국전자통신연구원 컴퓨터소프트웨어연구소 부장

관심분야 : 데이터베이스, 분산처리, 소프트웨어공학

임 기 욱

e-mail : rim@sunmoon.ac.kr

1977년 인하대학교 공과대학 전자공학과

1987년 한양대학교 전자계산학 석사

1994년 인하대학교 전자계산학 박사

1977년~1983년 한국전자기술연구소 선임연구원

1983년~1988년 한국전자통신연구소 시스템소프트웨어 연구실장

1988년~1989년 미 캘리포니아 주립대학(Irvine) 방문연구원

1989년~1996년 한국전자통신연구원 시스템연구부장 주전산기(타이컴) Ⅲ, Ⅳ개발 사업책임자

1997년~1999년 정보통신연구진흥원 정보기술전문위원

2001년~2003년 한국전자통신연구원 컴퓨터소프트웨어연구소장

2000년~현재 선문대학교 공과대학 지식정보산업공학과 교수

관심분야 : 실시간데이타베이스시스템, 운영체제, 시스템구조