1999年度 大韓電子工學會 秋季綜合學術大會 論文集 第 22 卷 第2號 99/11

SCI 연결망의 B-Link 인터페이스 회로 구현

한종석, 모상만, 기안도, 한우종
한국전자통신연구원 컴퓨터.소프트웨어연구소 하드웨어구조연구팀

Implementation of a B-Link Interface Logic for a SCI Interconnect

Jong-Seok Han, Sang-Man Moh, An-Do Ki, Woo-Jong Hahn
Hardware Architecture Research Team, ETRI-CSTL
jshan@computer.etri.re.kr

Abstract

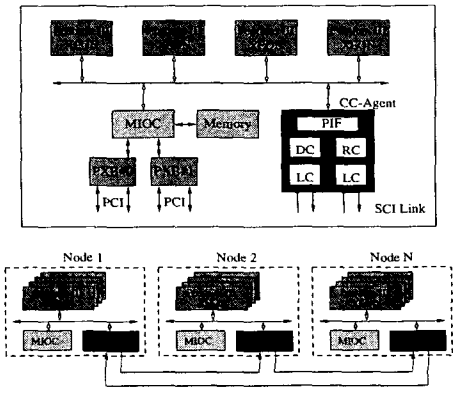
In this paper, we describe an implementation of the B-Link bus interface logic for a directory controller and a remote access cash controller in the SCI-based CC-NUMA multimedia server developed by ETRI. The CC-NUMA multimedia server is composed of a number of Pentium III SHV nodes and a SCI interconnection network. To communicate with remote nodes, each node has a CC-Agent which consists of a processor bus interface(PIF), a directory controller(DC), a remote access cash controller(RC), and two SCI link controllers(LCs).

The B-Link bus interface logic is developed for a directory controller and a remote access cash controller in order to communicate with a SCI link controller on a B-Link bus. It consists of a sending master controller, a receiving slave controller, and asynchronous data buffers. And it performs a self-arbitration, a data packet transmission, a queue allocation, an early termination, and a cut-through data path.

I. 서 론

CC-NUMA(cache coherent non-uniform memory access) 병렬 처리 시스템은 공유 메모리 구조의 SMP (symmetric multi-processor) 노드와 이를 연결하는 상호연결망으로 구성된다. CC-NUMA 병렬 처리 시스템은 공유 메모리 구조를 기반으로 SMP 노드간 캐쉬 일관성 유지 프로토콜을 제공하므로써 프로그램이 용이한 장점이 있다. 상용 CC-NUMA 시스템으로는 Data General 사의 AViiON 25000[1], Sequent 사의 NUMA-Q 2000[2], 그리고 SGI 사의 Origin 2000[3]등이 대표적이며, CC-NUMA 시스템의 상호연결망을 제공하는 Fujitsu System Technology 사의 SynfinityNUMA[4]등이 있다. 특히, Data General 사의 AViiON 25000 과 Sequent 사의 NUMA-Q 2000 은 SCI(scalable coherent interface) 상호연결망[5]을 기반으로 하는 대표적인 CC-NUMA 시스템이다. SCI 상호연결망은 IEEE 에서 표준으로 지정된 상호연결망으로서 linked-list 방식의 캐쉬 일관성 유지 프로토콜을 지원하며 다양한 구조의 연결 토폴로지를 제공한다.

현재 ETRI 에서 개발중인 멀티미디어 서버 시스템은 2 중 링(ring) 구조의 SCI 상호연결망을 기반으로 하는 CC-NUMA 시스템이다. 비록 SCI 상호연결망을 채택하고 있지만 linked-list 방식의 캐쉬 일관성 유지 프로토콜을 사용하지 않고 full-map 방식의 캐쉬 일관성 유지 프로토콜을 사용한다. <그림 1>은 CC-NUMA 멀티미디어 서버 시스템의 개략 구조를 보여준다. 각 SMP 노드는 4 개의 Pentium III Xeon 프로세서를 실장하고 있



<그림 1 : CC-NUMA 멀티미디어 서버 시스템>

으며 메모리 제어 및 PCI 버스 제어를 담당하는 MIOC(memory & IO controller)와 full-map 방식의 캐쉬 일관성 유지 프로토콜을 제공하는 CC-Agent 로 구성된다. CC-Agent 는 프로세서 버스 인터페이스 제어기, 디렉토리 제어기, 원격 캐쉬 제어기, 그리고 2 개의 SCI 링크 제어기로 구성된다. SCI 링크 제어기를 통하여 다수의 SMP 노드가 연결되므로써 하나의 CC-NUMA 시스템을 형성하게 된다.

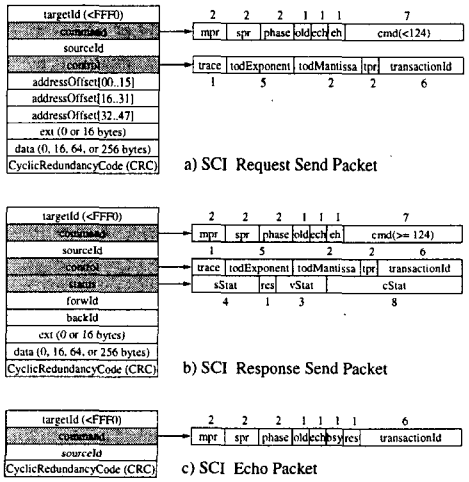
CC-Agent 내 SCI 링크 제어기의 한 쪽은 고속의 점대점 연결 링크를 제공하여 SCI 연결망을 구성하는데 사용되며, 다른 한 쪽은 B-Link 버스 인터페이스를 제공하여 CC-Agent 내 디렉토리 제어기 또는 원격 캐쉬 제어기와 링크 제어기간의 버스 연결을 구성하는데 사용된다. SCI 링크 제어기와 B-Link 버스 인터페이스는 Dolphin 사에서 개발하여 제공하는 제어기 칩과 버스 규격이다. CC-Agent 내 디렉토리 제어기와 원격 캐쉬 제어기는 B-Link 버스를 통해 링크 제어기와 데이터를 송수신하며, 이를 위해 내부에 B-Link 버스 규격을 만족하는 B-Link 버스 인터페이스 회로를 가져야 한다.

본 논문은 B-Link 버스 인터페이스 회로의 구현에 관한 것으로 1 장의 서론에 이어 2 장에서는 개략적인 SCI 규격과 B-Link 버스 규격을 소개하고 3 장에서는 B-Link 버스 인터페이스 회로의 구조 및 기능 구현에 대해 기술한다. 마지막으로 4 장에서는 구현 결과와 함께 결론을 맺고자 한다.

II. 연구 배경

1. Scalable Coherent Interface 규격

SCI 상호연결망은 IEEE 에서 표준으로 지정된 고속의 점대점 구조 상호연결망으로서 *linked-list* 방식의 캐쉬 일관성 유지 프로토콜을 지원하며 다양한 구조의 연결 토폴로지를 제공한다. 본 절에서는 주요 송수신 패킷 형태와 코딩 방법에 대해 간략히 언급하며 자세한 규격은 SCI 규격[5]을 참조하기 바란다.



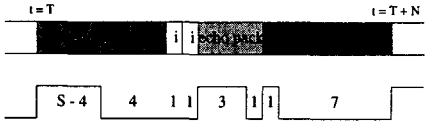
<그림 2 : SCI 규격에서 정의된 기본 패킷 형태>

1) 패킷 형태

SCI 패킷은 기본적으로 요청 전송(Request Send) 패킷, 응답 전송(Response Send) 패킷, 요청 전송 확인(Request Send Echo) 패킷, 그리고 응답 전송 확인(Response Send Echo) 패킷으로 구분된다. 이밖에 초기화 과정에서 사용되는 초기화(Init) 패킷과 전송 동기를 위해 사용되는 동기화(Sync) 패킷등이 있다. <그림 2>는 SCI 규격에서 정의된 기본 패킷 형태를 보여준다.

2) 패킷 인코딩/디코딩

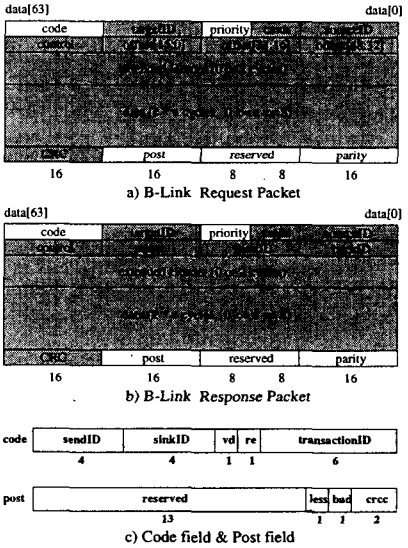
SCI 상호연결망의 물리 전송 경로는 16 비트이며 각 패킷은 16 비트의 연속적인 스트림으로 구성된다. 수신 동기를 위한 동기 비트와 패킷을 구분하기 위한 유효 비트가 데이터와 함께 전송된다. <그림 3>은 유효 비트를 이용하여 각 패킷을 구분하는 인코딩/디코딩 방법을 보여준다. 요청 패킷과 응답 패킷에 대한 구분은 각 패킷의 내부 정보에 의해 결정된다.



<그림 3 : 유효 비트를 이용한 패킷 코딩>

2. B-Link Bus Interface 규격

B-Link 버스는 SCI 인터페이스를 근간으로 Dolphin 사에서 정의한 버스 규격이다. SCI 링크 제어기의 한 쪽은 SCI 인터페이스를 제공하며, 다른 한 쪽은 B-Link 버스 인터페이스를 제공한다.



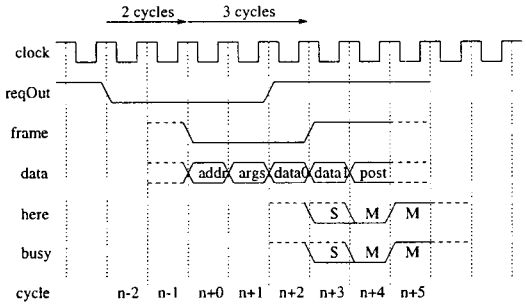
<그림 4 : B-Link Bus 규격에서 정의된 패킷 형태>

1) 패킷 형태

SCI 패킷과 달리 확인(Echo) 패킷은 지원하지 않고 버스 신호를 이용하여 요청 패킷이나 응답 패킷에 대한 수신을 확인한다. SCI 인터페이스가 16비트의 데이터 물리 전송 경로를 제공하는데 비해 B-Link 버스 인터페이스는 64비트의 데이터 물리 전송 경로를 제공한다. B-Link 패킷은 <그림 4>에 도시된 바와 같이 SCI 패킷을 포함하고 있으며 버스 중재 및 송수신에 필요한 추가 정보를 가지고 있다.

2) 패킷 코딩, 중재 및 전송

B-Link 버스에서는 공유 자원인 버스를 사용하기 위하여 <그림 5>에서 보는 바와 같이 송신측에서 먼저 버스 중재 요청 신호를 구동하여야 한다. 버스 중재에서 이긴 경우 송신측은 유효 신호와 데이터를 버스에 구동하여 전송을 시작하는데 구동 시점은 버스 중재 요청 신호 구동후 2 클럭후이며, 유효 신호가 해제된 후 2클럭째의 데이터까지 유효한 데이터이다. 수신측에서는 항상 유효 신호 구동후 3클럭후에 수신 확인 신호를 구동한다.



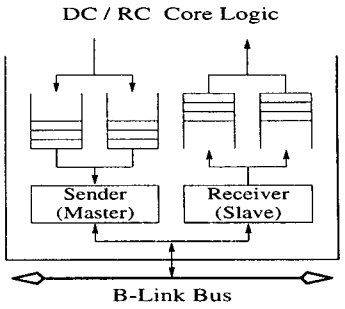
<그림 5 : B-Link 버스의 패킷 코딩, 중재 및 전송>

III. B-Link 버스 인터페이스 회로 구현

1. B-Link 버스 인터페이스 회로 구조

CC-NUMA 멀티미디어 서버 시스템의 디렉토리 제어기와 원격 캐쉬 제어기는 B-Link 버스를 통해 링크 제어기와 데이터를 송수신한다. B-Link 버스 인터페이스 회로는 상기 디렉토리 제어기와 원격 캐쉬 제어기내에 위치하여 해당 제어기에서 필요로 하는 데이터를 B-Link 버스 프로토콜에 맞게 송수신하는 역할을 담당한다. B-Link 버스 인터페이스 회로는 버스의 중재와 데이터 전송을 수행하는 송신부(sender), 링크 제어기로부터 데이터를 수신하는 수신부(receiver), 디렉토리 제어기 코아 또는 원격 캐쉬 제어기 코아와 데이터를 주고 받기 위한 비동기 데이터 버퍼(asynchronous data buffer)등으로 구성된다(그림 6).

비동기 데이터 버퍼는 송신 및 수신 버퍼가 각각 이중화되어 있어 연속으로 송신 또는 수신되는 데이터 패킷을 처리할 수 있다.



<그림 6 : B-Link 버스 인터페이스 회로 구조>

2. B-Link 버스 인터페이스 회로 주요 기능

1) 중재 및 전송

B-Link 버스는 중재를 위해 기본적으로 1개의 RequestOut 신호와 8개의 RequestIn 신호를 제공한다. 1개의 RequestOut 신호는 B-Link 버스상에 존재하는 각 디바이스의 RequestIn 신호중 하나에 연결되며, 이 신호는 모두 물리적으로 동일한 위치에 존재한다. B-Link 버스 인터페이스 회로는 구동된 RequestIn 신호를 감지하여 라운드 로빈 방식의 분산 자가 중재(distributed self-arbitration)를 수행한다.

중재에서 이긴 경우 송신 데이터 버퍼에 저장된 데이터의 헤더 정보를 이용하여 B-Link 버스의 목적지 주소를 생성하며, 패킷의 프레임 및 크기를 결정하고, 패리티를 생성하여 B-Link 전송 프로토콜에 따라 해당 데이터 패킷을 전송한다. 일정 클럭후 수신 확인 신호를 감지하여 재전송 요구일 경우 다시 중재를 거쳐 데이터 패킷을 재전송하며, 해당 수신 노드가 없을 경우 전송을 중단하고 에러 신호를 디렉토리 제어기 코아 또는 원격 캐쉬 제어기 코아로 구동한다.

B-Link 버스 인터페이스 회로의 수신부는 B-Link 버스에 구동된 데이터 패킷을 디코딩하여 목적지 주소를 인지하고, 데이터 패킷을 수신 데이터 버퍼에 저장하며, 내부 수신 데이터 버퍼의 여유가 없을 경우 재전송을 요구한다. B-Link 버스는 패킷 에러에 대한 재전송을 허용하지 않기 때문에 수신 확인 신호를 구동하여 송신측에 패킷 수신 상황을 알리고 패리티를 검사하여 이상이 있을 경우 디렉토리 제어기 코아 또는 원격 캐쉬 제어기 코아로 내부 패킷 에러 신호를 구동한다.

2) 큐 배정 프로토콜

공정한 중재 보장과 기아 현상 방지는 버스 프로토콜에서 제공하여야 할 중요한 요소이다. 공정한 중재를 통해 전송을 시도하지만 특정한 버스 상황에 따라 번번이 재전송을 할 경우 기아 현상과 유사한 상황이 발생할 수 있기 때문에 보다 합리적인 전송을 위해 큐 배정(queue allocation) 프로토콜을 제공한다. 재전송 요구를 받은 송신측은 계속해서 데이터 패킷을 재전송하게 되는데, 이 때 패킷의 헤더에 재전송 패킷임을 알리는 정보를 실어 전송하므로써 수신측에서 다른 패킷에 우선해 패킷을 수신할 수 있도록 한다.

3) 전송 조기 중단

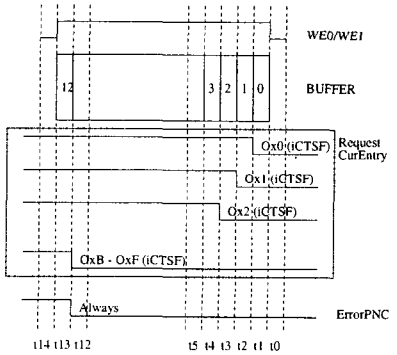
B-Link 버스는 최대 256 바이트 크기의 데이터를 전송할 수 있다. 수신측에서 수신 데이터 버퍼의 여유가 없을 경우 바로 패킷 재전송요구를 하게 되는데 송신측에서는 이를 인지하고도 256 바이트 크기의 데이터 패킷을 버스에 계속 실어 보내야 하는 문제가 발생한다. 전송 조기 중단 기능은 송신측에서 이를 인지할 경우 바로 데이터 패킷 전송을 중단하므로써 버스 대역폭 낭비를 방지하고 보다 빠른 전송을 수행할 수 있다. 실제, 본 논문의 B-Link 버스 인터페이스 회로에서는 최대 64 바이트 크기의 데이터 전송이 가능하도록 구현되어 기대 효과가 크지는 않지만 효율적인 버스 대역폭을 제공하기 위하여 64 바이트 전송에 대한 조기 전송 중단 기능을 제공한다.

4) 수신 확인 신호 복구

수신 확인 신호는 pull-up 저항을 사용하여 비활성 상태에서 고전압 상태로 복원(restore)되는 low-active 출력 신호이다. 수신측에서 이 신호를 활성화하여 구동한 후 비활성화시킬 때 복원까지 시간이 오래 걸릴 경우 문제가 발생할 수 있다. 수신측에서 구동된 수신 확인 신호를 바로 다음 클럭에 송신측에서 거두어 들이므로서 이러한 문제를 해결할 수 있다. B-Link 버스 인터페이스 회로의 송신부에서 수신 확인 신호 복구 기능을 수행한다.

5) 비동기 데이터 버퍼 및 Cut-through 전송

B-Link 버스 인터페이스 회로는 디렉토리 제어기 코아 또는 원격 캐쉬 제어기 코아와 데이터를 주고 받기 위하여 비동기 송수신 데이터 버퍼 인터페이스를 제공한다. 비동기 데이터 버퍼는 B-Link 버스 인터페이스 회로와 서로 다른 주파수로 동작하는 디렉토리 제어기 코아 또는 원격 캐쉬 제어기 코아를 지원하기 위해 구현되었으며, Cut-through 전송 방식을 제공하여 Store-and-forward 방식의 비동기 인터페이스보다 적은 전송 지연시간을 갖도록 구현하였다. 다양한 종류의 비동기 동작 주파수를 지원하기 위하여 내부에 제어 상태 레지스터를 두어 가변적으로 제어할 수 있도록 구현하였다.



<그림 7 : 비동기 인터페이스의 요청 신호 구동 시점>

<그림 7>은 B-Link 버스에서 수신된 데이터를 비동기 수신 데이터 버퍼에 저장하면서 동시에 디렉토리 제어기 코아 또는 원격 캐쉬 제어기 코아로 데이터 처리 요청 신호를 구동하는 시점을 보여준다. 요청 신호 구동 시점은 CTSF 레지스터의 값에 따라 달라진다.

B-Link 버스 인터페이스 회로의 동작 속도가 디렉토리 제어기 코아 또는 원격 캐쉬 제어기 코아의 동작 속도와 같거나 빠를 경우 t1 클럭에서 요청 신호를 구동하며, 늦을 경우 동작 속도 차에 따라 t2 에서 t12 까지 CTSF 레지스터의 값에 따라 요청 신호를 구동한다.

V. 결 론

B-Link 버스 인터페이스 회로는 CC-NUMA 멀티미디어 서버 시스템내 CC-Agent 를 구성하는 디렉토리 제어기와 원격 캐쉬 제어기의 내부 회로로서 B-Link 버스를 통해 SCI 링크 제어기와 데이터를 송수신하기 위해 구현되었다. 본 회로는 SCI 규격을 기반으로 하는 B-Link 버스 프로토콜에 따라 SCI 링크 제어기와 데이터 패킷을 송수신할 수 있고, 비동기 송수신 데이터 버퍼를 통하여 다른 주파수로 동작하는 디렉토리 제어기 코아 또는 원격 캐쉬 제어기 코아와 데이터를 주고 받을 수 있다.

현대 0.35 um standard cell 라이브러리를 사용하여 로직 합성과 시뮬레이션을 수행하였다. 최대 100 MHz 에서 동작하며, 내부 PLL on/off 기능을 제공하고, Dolphin 사의 LC2 및 LC3 링크 제어기와 인터페이스가 가능하도록 구현하였다.

참고 문헌

[1] Data General, "SCI Interconnect Chipset and Adapter: Building Large Scale Enterprise Servers with Pentium II Xeon SHV Nodes", http://www.dg.com/about/html/sci_interconnect_chipset_and_a.html
[2] Sequent, "NUMA-Q 2000", http://www.sequent.com/products/highend_srv/
[3] Silicon Graphics Inc., "SGI Origin ccNUMA Architecture", http://www.sgi.com/origin/numa.html
[4] Fujitsu System Technologies, "Synfinity NUM", http://www.fjst.com/products/synfinitynuma/
[5] IEEE Std 1596-1992, "IEEE Standard for Scalable Coherent Interface (SCI)"