論文 94-19-7-9

계층 클러스터 구조를 위한 캐쉬 일관성 프로토콜의 설계 및 구현

正會員 朴 信 民* 正會員 崔 昌 勳** 正會員 金 聖 天**

A Design and Implementation of Cache Coherence Protocol for Hierarchical Cluster Architecture

Sin Min Park*, Chang Hoon Choi**, Sung Chun Kim** *Regular Members*

요 약

본 논문에서는 계층 버스 구조를 기반으로 하는 계층 클러스터 다중프로세서 시스템(hierarchical cluster multiprocessor system)을 제안하고, 이 구조에 효율적인 캐쉬 일관성 프로토콜을 설계 및 구현한다. 계층 클러스터 구조는 클러스터 수의 증가에 따라 버스 계층을 추가함으로써 기존의 단일 버스 시스템에서의 병목 현상(bottleneck)을 없애주어 시스템 확장이 용이하게 된다. 제안된 캐쉬 프로토콜은 일반적인 N-레벨 ($N \geq 2$) 계층 클러스터 구조에 적용할 수 있도록 설계되었다. 이를 시스템 버스에 구현하기 위하여 기존의 팬디드 프로토콜을 확장하였고 캐쉬 일관성 동작들을 확장된 팬디드 프로토콜상에서 설명한다.

Abstract

In this paper, a hierarchical cluster multiprocessor system based on a hierarchical bus system is proposed and its cache coherency protocol is designed and implemented. The hierarchical cluster architecture aims at eliminating the system bottleneck of the existing single bus system by adding a hierarchy of buses as the number of clusters is increased. Therefore the system is easy to scale up to a large number of processors. The proposed cache protocol is designed to be adapted to the general N-level ($N \geq 2$) hierarchical cluster architecture. The original pended protocol is extended to implement the cache protocol on the system bus and cache coherency operations for this protocol are explained.

* 한국통신 통신망연구소
Telecom. Net. Research Lab, Korea Telecommunication
** 서강대학교 전자계산학과
Dept. of Computer Science, Sogang University

論文番號 : 9446
接受日字 : 1994年 2月 15日

I. 서 론

다중프로세서 시스템의 설계에 있어서 중요한 문제는 시스템에 존재하는 여러 모듈들간에 빠른 통신

을 지원해 줄 수 있는 충분한 대역폭의 상호 연결망(interconnection network)의 구성이다. 또한 프로세서의 빈번한 공유 메모리 참조 요청으로 발생하는 시스템의 성능 저하를 막기 위해 프로세서와 메모리 사이에 캐쉬(cache)를 두어야 한다. 이런 캐쉬는 프로세서 싸이클과 메모리 싸이클 사이의 속도 차를 보완하고 상호 연결망의 트래픽 부하를 줄일 수 있는 잇점을 주지만, 같은 메모리 블럭의 복사본이 여러 캐쉬에 존재할 수 있기 때문에 복사본들 사이의 일관성을 유지해야 하는 캐쉬 일관성 문제(cache coherency problem)가 발생한다. 이를 해결하기 위한 여러가지 형태의 캐쉬 일관성 프로토콜이 개발 되었는데, 이 기법은 캐쉬와 메모리 모듈 사이의 통신 프로토콜로 구성되어 있다[1-5].

최근에는 공유 버스의 한계와 MIN 구조의 문제점을 보완하기 위하여, 변형된 상호 연결망을 사용하여 캐쉬 일관성 문제를 쉽게 해결하고 시스템의 확장성을 최대로 보장해 줄 수 있는 캐쉬 일관성 네트워크 구조(cache coherency network architecture)에 대한 연구가 활발히 진행되고 있다. 이런 구조들은 구현상의 복잡도가 심하지 않고, 상호 연결망의 대역폭의 한계로 인하여 시스템이 확장에 제약을 받지 않게 하기 위하여 일반적으로 계층 버스 구조를 사용한다. 이러한 계층 버스 구조로는 Wilson이 제안한 계층 캐쉬/버스 구조와 클러스터(cluster) 구조 [6], Goodman과 Woest가 제안한 Wisconsin Multicube[7], 스탠포드 대학에서 현재 개발중에 있는 DASH(Directory Architecture for SHared memory) 다중프로세서[8], 그리고 스웨덴 국립 과학 연구소에서 제안한 계층 버스로 구성된 DDM(Data Diffusion Machine) 시스템[9] 등이 있다.

이들 구조 중 Wilson이 제안한 계층 캐쉬/버스 구조는 프로세서와 메모리 사이에 다단계의 버스와 캐쉬 계층을 두어, 이들을 트리 형태로 연결한 잘 알려진 구조의 하나이다. 이 구조는 버스와 캐쉬 계층의 추가만으로 시스템을 쉽게 확장할 수 있지만, 시스템의 크기가 커지면 여러개의 레벨 버스를 거쳐 메모리를 엑세스하게 되므로 평균 메모리 엑세스 지연 시간이 길어지고, 또한 메모리 요청과 무효화 방송등으로 인한 트래픽이 상위 레벨 버스로 집중되어 최상위 레벨 버스는 시스템 병목(system bottleneck)을 형성한다. 이와 같은 최상위 레벨 버스의 한정된 대역폭은 계층 캐쉬/버스 구조의 확장에 제한 요인으로 작용한다[10, 11].

또한 Wilson은 계층 캐쉬/버스 구조의 단점을 보완한 클러스터 구조를 제안하였는데, 이 구조는 프로세서들의 그룹으로 형성된 클러스터안으로 공유 메모리 모듈을 분산시켜 둠으로써, 전역 버스(global bus)의 트래픽과 메모리 응답 지연을 줄이는 장점을 갖는다. 그러나 단일 공유 버스를 사용하여 클러스터들을 연결하였기 때문에 시스템 확장에 한계를 갖는다.

본 논문에서는 위와 같은 Wilson의 두가지 구조의 단점을 보완하는 계층 클러스터 구조를 제안한다. 이 구조는 계층 버스를 사용하여 클러스터들을 연결하는 형태이며 공유 메모리 모듈들을 각 클러스터로 분산시킴으로써 빠른 메모리 엑세스가 가능하고, 또한 클러스터 수의 증가에 따라 버스 계층의 추가로 시스템의 확장이 용이하고, 각 레벨 버스에 균등한 트래픽 부하를 주어 전체 버스 대역폭을 최대화할 수 있는 장점을 갖는다. 그리고 제안된 구조에서 효율적으로 동작하는 캐쉬 일관성 프로토콜을 제안하고, 이의 구현을 위하여 확장된 팬디드 프로토콜을 설계하고, 시스템의 성능 평가를 위하여 시뮬레이션을 수행한다.

본 논문의 구성은 다음과 같다. 제1장의 서론에 이어 제2장은 일반적인 계층 버스 구조의 특성을 나타내고, 제3장은 본 논문에서 제안한 계층 클러스터 구조의 특성 및 각 시스템 모듈의 구성을 설명하고, 제4장은 계층 클러스터 구조를 위한 캐쉬 일관성 프로토콜을 설계한다. 그리고 5장에서는 확장된 팬디드 프로토콜을 설계하여 제안된 캐쉬 일관성 동작이 버스상에서 어떻게 수행되는지를 설명하고, SLAM II 시뮬레이션 언어를 사용하여 시뮬레이션을 수행한다. 끝으로 6장에서 결론을 맺는다.

Ⅱ. 일반적인 계층 버스 구조

2.1 계층 캐쉬/버스 구조

Wilson이 제안한 계층 버스/캐쉬 구조의 다중프로세서 시스템은 〈그림 2-1〉과 같이 구성된다. 이 구조는 프로세서에 직접 연결된 작지만 빠른 속도의 하위 레벨 캐쉬(lower-level cache)들이 단일 공유 버스에 연결되어 하나의 클러스터를 구성한다. 이 클러스터들이 모여 약간 느리지만 커다란 크기의 상위 레벨 캐쉬(higher-level cache)에 연결되며, 이 상위 레벨 캐쉬들은 전역 버스를 통해서 메모리와 연결된다[6].

계층 캐쉬/버스 구조는 다음과 같은 몇가지 문제점을 갖는다. 첫번째로, MLI 특성을 보장하기 위해서는 상위 레벨 캐쉬를 매우 크게 구현해야 한다는 점

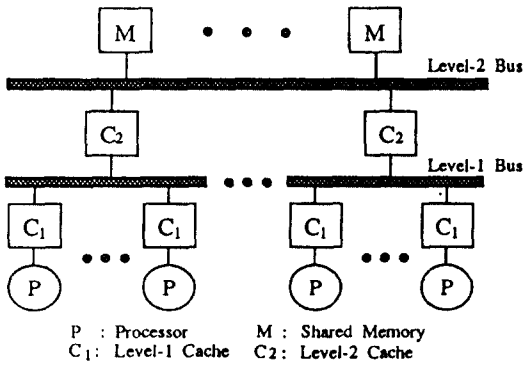


그림 2-1. Wilson의 계층 버스/캐쉬 구조.

이다. 이것은 많은 레벨이 필요한 대형 시스템에서는 상위 레벨로 갈수록 캐쉬 메모리의 부담이 크고, 이런 캐쉬를 구현하는 느린 속도의 DRAM으로 인하여 메모리 요청에 대한 실패시 메모리 접근 시간이 길어진다. 두번째로, 메모리가 최상위 레벨 버스에 연결되어 있기 때문에 메모리 엑세스 요구와 무효화 방송과 같은 캐쉬 일관성 요청들이 최상위 레벨 버스로 집중되어서 시스템 병목 현상이 발생한다. 세번째로, 최상위 레벨 버스의 트래픽 과부하로 인하여 시스템 확장성의 제약을 받는다.

2.2 클러스터 구조

계층 캐쉬/구조와 함께 Wilson이 제안한 클러스터 구조는 〈그림 2-2〉와 같이 구성된다. 이 구조는 Encore 컴퓨터 사이에 설계한 Ultramax의 구조로 사용되었으며, 한 클러스터는 CPU 보드 8장 (2-CPU/1 board), 메모리 보드 8장, 레벨-2 캐쉬 보드 1장, 인터페이스 카드 1장으로 구성되며, 최대 8개의 클러스터가 한개의 전역 공유 버스를 통하여 연결된다. 전역 버스와 클러스터내의 버스는 나노 버스(nanobus)를 사용한다[12].

이런 구조는 메모리를 클러스터 내에 둠으로써 프로그램 수행시에 큰 지역성을 제공하여 빠른 메모리 엑세스가 가능하다. 그러나 클러스터들을 연결하는 전역 버스의 대역폭의 한계로 확장에 제약을 받는다. 이에 대한 근거는 Wilson의 시뮬레이션 결과에서 볼 수 있는데, 8개의 클러스터로 구성된 시스템에 대해 시 시뮬레이션을 수행한 결과, 전역 버스의 사용률이 60%를 넘어서 버스가 포화 상태에 있음을 보인다[6]. 따라서 시스템의 확장을 위해서는 클러스터들을 연결하는 버스 구조를 보다 확장 가능한 구조로 바꿀 필요가 있다.

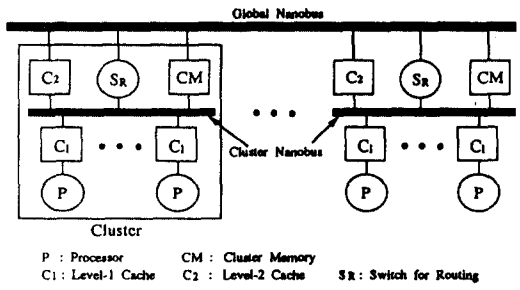


그림 2-2. Wilson의 클러스터 구조.

Ⅲ. 계층 클러스터 구조

3.1 계층 클러스터 구조 및 특징

본 장에서는 공유 메모리를 사용하는 확장 가능한 다중프로세서 시스템을 위한 계층 클러스터 구조를 제안하고, 이의 구현 방안을 제시한다. 이 구조는 여러개의 클러스터들이 계층 버스를 통하여 상호 연결되는 형태로서 기존의 클러스터 구조가 갖는 시스템 확장성의 한계를 해결한다.

〈그림 3-1〉은 3-레벨의 계층 버스를 사용한 계층 클러스터 구조로서, 여러 클러스터들을 연결하기 위하여 두개의 레벨-2 버스와 한개의 레벨-3 버스를 사용하였다. 각 클러스터는 프로세서와 레벨-1 캐쉬(프로세서 캐쉬 : PC), 레벨-2 캐쉬(클러스터 캐쉬 : CC), S_R (Switch for Routing), 지역 메모리(클러스터 메모리 : CM)로 구성되고, 이런 모듈들은 단일 공유 버스인 레벨-1 버스를 통하여 상호 연결된다.

레벨-2 캐쉬는 자신의 하위 레벨 캐쉬 내용을 모두 포함하는 MLI특성을 갖기 때문에, 입출력(I/O)과 다른 클러스터의 캐쉬 일관성 간섭으로부터 하위 캐쉬들을 보호해 주며, 하위 레벨 캐쉬의 원격 요청(remote request)을 상위 레벨 버스로 전달해 주는 역할을 한다[13]. S_R은 외부 클러스터가 지역 메모리를 요청할 경우, 이 블럭을 지역 메모리에서 가져와 요청 클러스터로 보내주고 그에 대한 정보를 S_R 메모리에 기록한다. S_R 메모리는 외부 클러스터에 존재하는 자신의 지역 메모리 블럭의 상태를 유지하는 디렉토리로 사용된다. 또한 클러스터간의 효율적인 통신을 위해서

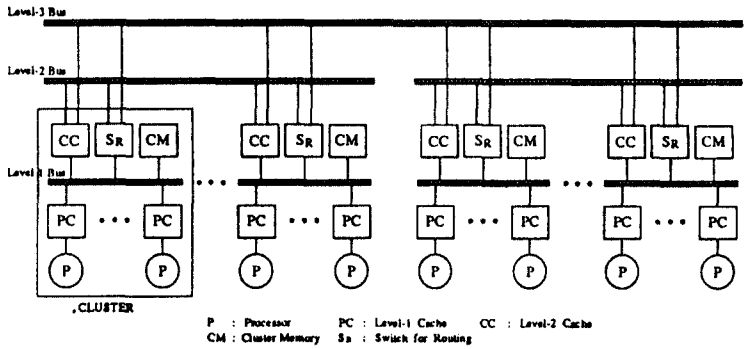


그림 3-1. 3-레벨 계층 클러스터 구조의 다중프로세서 시스템.

S_R 메모리에는 블럭이 공유된 최상위 버스 레벨(Shared Bus Level : SBL)을 기록하여, 그 블럭에 대한 프로세서의 요청을 레벨-2 캐쉬가 효과적으로 처리하게 된다.

3.2 계층 클러스터 구조의 장점

계층 클러스터 구조는 기존의 계층 버스 구조와 비교하여 다음과 같은 장점을 갖는다.

3.2.1 평균 메모리 엑세스 지연 시간(average memory access delay time)

계층 클러스터 구조에서는 공유 메모리 모듈이 분산되어 각 클러스터안에 존재하므로 클러스터내 메모리를 엑세스하는 지역 요청은 빠르게 처리되며, 다른 클러스터 메모리를 엑세스하는 원격 요청은 상대적으로 느리게 처리된다. 따라서 프로세서의 메모리 참조가 해당 클러스터 메모리에서 제공될 확률이 높을수록 평균 메모리 엑세스 지연 시간은 줄어든다.

3.2.2 상위 레벨 버스의 트래픽

본 논문에서 제안하는 구조에서는 클러스터의 수를 늘리고자 할때 버스 계층을 추가할 수 있고, 원격 요청을 여러 상위 레벨 버스로 분산시킬 수 있으므로 상위 레벨 버스의 트래픽을 감소시킨다.

3.2.3 사용되는 캐쉬 메모리의 양

본 구조에서는 캐쉬 계층의 증가없이 단지 버스 계층만 추가하여 구성하므로 캐쉬 메모리의 사용량을 늘리지 않는다. 또한 S_R 메모리는 실질적인 데이타를 저장하지 않으므로 적은 양의 메모리로 구현할 수 있다. 따라서 모든 캐쉬 메모리를 고속의 SRAM을 사용하여 구성하는 것이 가능하여 빠른 엑세스를 할 수 있다.

3.2.4 시스템의 확장성

기존의 클러스터 구조의 제약점인 시스템 확장성의 한계를 계층 클러스터 구조에서는 버스 계층을 추가하여 해결할 수 있다. 그러나 버스 계층이 하나씩 추가될 때마다 레벨-2 캐쉬와 S_R 각각에 버스 인터페이스 회로와 스누퍼를 추가해야 하므로 설계의 복잡도가 증가되는 문제를 갖는다.

3.3 시스템 모듈의 구성도

계층 클러스터 구조에서 한 클러스터는 〈그림 3-2〉와 같이 프로세서 보드, 클러스터 캐쉬 보드, S_R 보드, 메모리 보드 등 4가지 종류의 보드로 구성된다. 그리고 이런 각 보드들은 여러개의 독립적인 모듈들로 나뉘어 구성된다.

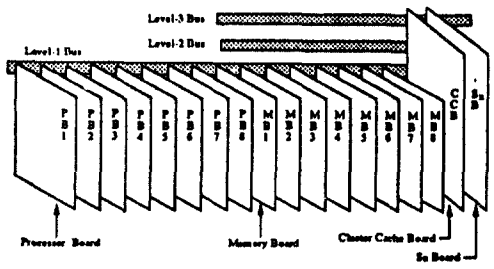


그림 3-2. 3-레벨 계층 클러스터 구조에서의 클러스터 구성도.

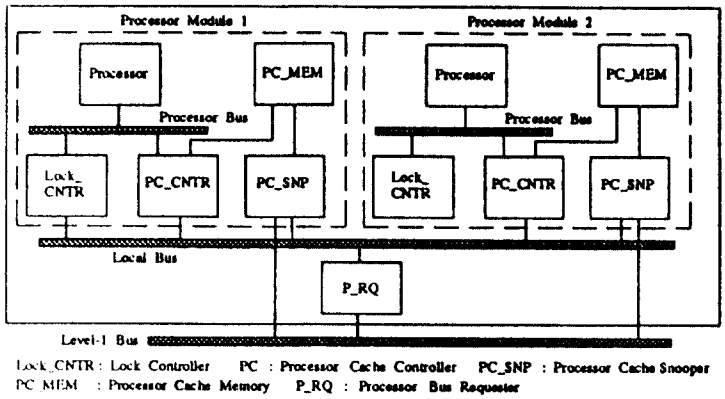


그림 3-3. 프로세서 보드의 구성도.

3.3.1 프로세서 보드의 구성과 동작

한 프로세서 보드에는 〈그림 3-3〉과 같이 두개의 프로세서 모듈이 장착되며, 이들은 하나의 버스 요청기(P_RQ)를 공유하여 사용한다. 각 프로세서 모듈은 프로세서, 캐쉬, Lock 제어기(Lock_CNTR)로 구성된다.

3.3.2 클러스터 캐쉬 보드의 구성과 기능

클러스터 캐쉬 보드는 〈그림 3-4〉와 같이 스누프 제어기(CC_SNPi, i = 1, 2, 3), 버스 요청기(CC_RQi), 캐쉬 메모리(CC_MEM), 캐쉬 제어기(CC_CNTR)로 구성된다. 각 레벨 버스를 감시하는 스누프 제어기는 버스로 데이타를 구동할 수 있는 데이타 구동 제어기(Data Drive Controller : DDC)를 갖고, 특히 CC_SNP1은 하위 레벨-1 캐쉬로 전송되는 블럭을 래치하기 위해 데이타 래치 제어기(Data Latch Controller : DLC)를 포함한다.

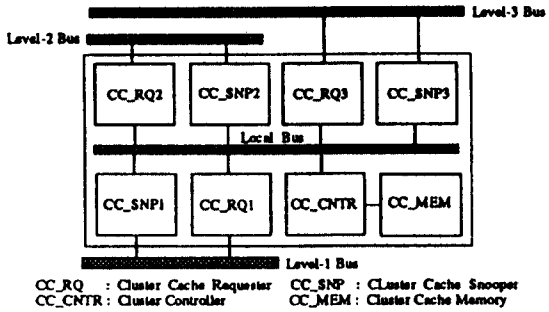


그림 3-4. 클러스터 캐쉬 보드의 구성도.

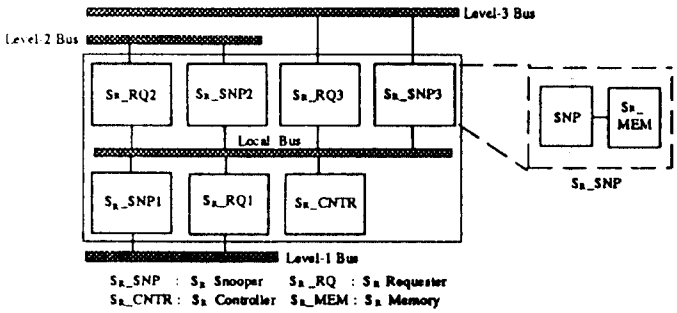


그림 3-5. S_R 보드의 구성도.

3.3.3 S_R 보드의 구성과 기능

〈그림 3-5〉처럼 S_R 보드는 S_R 메모리(S_R_MEM), S_R 제어기(S_R_CNTR), 스누프 제어기(S_R_SNPi), 버스 요청기(S_R_RQi)로 구성된다. 스누퍼들은 버스를 감시하여 블럭의 전송 형태에 따라 상태와 SBL 값을 변경한다. 레벨-2 이상의 스누퍼들은 다른 클러스터로 부터의 요청을 S_R_RQ1에게 보내는 일을 담당한다.

3.34 메모리 보드의 구성과 동작

메모리 보드는 〈그림 3-6〉과 같이 2-way 인터리브드(interleaved) 방식으로 구성된 메모리 모듈과 버스 응답기(MEM_RP)로 구성된다. 그리고 버스 동작의 속도를 향상시키기 위해 MEM_RP는 파이프라인화된 요청 버퍼 레지스터들을 사용하여 메모리 제어기가 이전의 요청을 수행 중 일지라도 버스상의 요청을 거절하지 않고 요청 버퍼 레지스터로 보내 메모리가 해제될 때 서비스하게 한다.

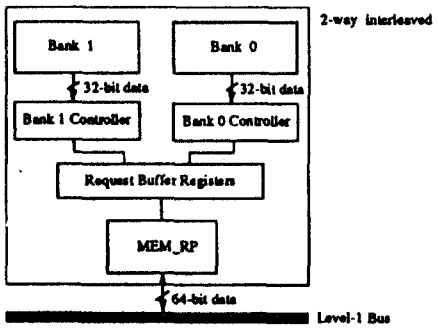


그림 3-6. 메모리 보드의 구성도.

Ⅳ. 계층 클러스터 구조를 위한 캐쉬 일관성 프로토콜

이 장에서는 공유 메모리를 기반으로 하는 계층 클러스터 다중프로세서를 위한 캐쉬 일관성 프로토콜을 설명한다. 본 논문에서 제안한 캐쉬 프로토콜은 기록-무효화(write-invalidation) 방법과 기록-복귀(write-back) 방법을 기본으로 하여 설계되었고, 블럭의 마스터쉽(mastership) 개념을 사용한다.

4.1 각 모듈의 캐쉬 블럭의 상태 정의

4.1.1 레벨-1 캐쉬 블럭의 상태

· VALID_SHARED(VS) : 클러스터내 다른 레벨-1 캐쉬와 레벨-2 캐쉬에 그 블럭이 공유되어 있는 상태로, 프로세서의 읽기 요청은 지연없이 즉시 처리되지만, 기록 요청은 다른 공유 블럭들을 무효화 시키고 마스터쉽을 획득한 후에 수행된다. 블럭의 대치시 기록-복귀될 필요는 없다.

· VALID_MASTER(VM) : 수정되거나 수정되지 않은 상태로 시스템에 존재하는 유일한 블럭으로 프로세서의 읽기와 기록 요청은 지연없이 즉시 수행된다. 그러나 레벨-1 버스를 통한 다른 프로세서로 부터의 기록 요청에 대해서는 블럭을 공급하고 상태는 I로 바뀐다. 레벨-1 버스의 읽기 요청시에는 블럭을 공급하고 상태는 VS로 바뀌고, 블럭의 마스터쉽은 레벨-2 캐쉬로 이동된다. 블럭의 대치시에는 메모리로 기록-복귀가 수행된다.

· INVALID (I) : 무효화된 상태이다.

4.1.2 레벨-2 캐쉬 블럭의 상태

· VALID_SHARED(VS) : 한개 이상의 클러스터에 유효한 블럭이 존재하는 경우이다. 클러스터내 레벨-1 캐쉬나 다른 클러스터 캐쉬의 읽기 요청에 대해서, 그 블럭은 즉시 공급되지만, 기록 요청은 먼저 다른 공유 블럭들을 무효화를 시킨후에 처리된다. 블럭의 대치시에는 해당 S_R로 대치됨을 알린다.

· VALID_MASTER(VM) : 수정되거나 수정되지 않은 상태로 단 한개의 클러스터에만 존재하는 블럭으로 레벨-2 캐쉬가 그 블럭의 마스터쉽을 갖는다. 클러스터내 레벨-1 캐쉬들은 VS 상태로 그 블럭을 공유할 수 있다. 하위 레벨-1 캐쉬의 읽기 요청에 대해서 블럭은 즉시 공급되지만 기록 요청시에는 블럭의 마스터쉽은 하위 레벨-1 캐쉬로 옮겨가고, 레벨-2 캐쉬는 DO 상태로 바뀐다. 다른 클러스터로 부터의 읽기 요청은 블럭을 공급하고 상태를 VS로 바뀌고 기록요청은 먼저 하위 레벨-1 캐쉬로 무효화 방송을 한 후 블럭을 공급하고 상태를 I로 바꾼다. 대치시에는 메모리로 기록-복귀하고 하위 레벨 캐쉬들을 무효화 시킨다.

· DESCENDANT_OWNED(DO) : 클러스터내 한 레벨-1 캐쉬가 블럭의 마스터쉽을 갖는 경우이다. DO는 외부 클러스터의 요청시에 그 블럭의 마스터쉽을 갖는 레벨-1 캐쉬를 가리키는 포인터로 작용하여 엑세스 경로를 제공한다. DO 상태의 블럭은 실제로 데이타를 저장하지 않는다.

· INVALID(I) : 무효화된 상태이다.

4.1.3 S_R 메모리 블럭의 상태

· SHARED(S) : 다른 클러스터의 레벨-2 캐쉬에 VS 상태로 블럭이 존재한다.

· DIRTY(D) : 다른 클러스터의 레벨-2 캐쉬에 DO, VM 상태로 그 블럭이 있는 경우이다.

4.2 캐쉬 프로토콜의 상태 전이도

4.2.1 레벨-1 캐쉬 블럭의 상태 전이도

레벨-1 캐쉬 블럭의 상태 전이도는 〈그림 4-1〉과 같고, 요청의 형태는 프로세서의 요청과 버스상의 요청으로 나뉘어진다. 캐쉬에 없는 블럭은 I 상태와 같이 취급된다.

4.2.2 레벨-2 캐쉬 블럭의 상태 전이도

레벨-2 캐쉬 블럭은 〈그림 4-2〉와 같이 상태 전이를 하며, 요청의 형태는 레벨-1 캐쉬의 버스 요청과 다른 클러스터로 부터의 요청으로 분류된다. 레벨-1 캐쉬의 요청을 S_R의 요청과 구분하기 위해서 L1CR, L1CW 처럼 표기한다.

4.2.3 S_R 메모리 블럭의 상태 전이도

S_R 메모리에 있는 블럭의 상태 전이는 〈그림 4-3〉과 같으며, S_R은 자신의 클러스터 메모리에 대한 요청만 처리한다.

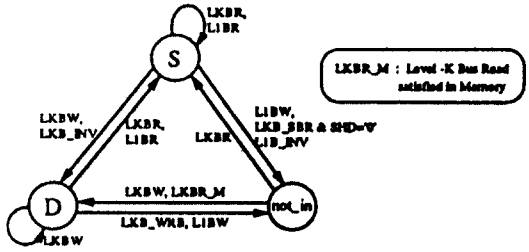


그림 4-3. S_R 메모리 블럭의 상태 전이도.

4.3 캐쉬 프로토콜의 동작

제안된 캐쉬 일관성 프로토콜에서 프로세서가 레벨-1 캐쉬를 엑세스할때 발생할 수 있는 동작들은 읽기 적중(Read Hit), 읽기 실패(Read Miss), 기록 적

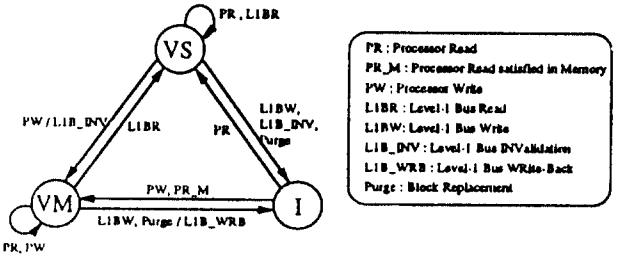


그림 4-1. 레벨-1 캐쉬 블럭의 상태 전이도.

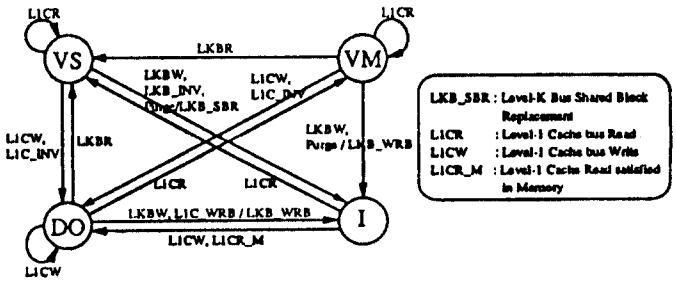


그림 4-2. 레벨-2 캐쉬 블럭의 상태 전이도.

중(Write Hit), 기록 실패(Write Miss), 블럭 대치(Block Replcement)로 나누어진다. 이 절에서는 레벨-1 캐쉬, 레벨-2 캐쉬, S_R이 각 동작에 대하여 어떻게 상호 관련되어 동작하는지를 설명한다.

4.3.1 사용되는 용어 및 상태 라인(Status Line)

캐쉬 프로토콜의 설명을 간편화하기 위하여 다음과 같은 용어와 상태 라인을 정의한다. 프로세서는 주소가 K인 블럭(블럭 K)을 요청한다고 가정하자.

- B[K] : 블럭 K를 포함하는 메모리 모듈과 연결된 최하위 버스 번호
- SBL[i] : S_R 메모리의 블럭 i가 다른 클러스터와 공유된 버스 번호
- DIRTY : 블럭이 VM 상태로 존재함을 알리는 버스상의 신호
- SHD : 레벨-2 캐쉬가 블럭이 VS 상태로 존재함을 알리는 버스상의 신호
- RC_DIRTY : S_R이 구동하는 신호로 다른 클러스터에 블럭이 VM 상태로 있음을 알리는 레벨-1 버스상의 신호
- RC_SHD : S_R이 구동하는 신호로 블럭이 다른 클러스터에 공유되어 있음을 알리는 레벨-1 버스상의 신호

4.3.2 읽기 적중

레벨-1 캐쉬가 해당 블럭을 공급하여 주며 상태의 변화는 없다.

4.3.3 읽기 실패

프로세서가 요청한 블럭 K가 레벨-1 캐쉬에 없을 경우, 레벨-1 캐쉬는 버스로 읽기 요청(Read For Read)을 보낸다. 이때 요청에 응답을 해주는 모듈은 레벨-2 캐쉬의 상태와 S_R 메모리의 상태에 따라 결정된다. 요청한 블럭이 레벨-1 버스로 전송될때, 레벨-2 캐쉬도 블럭을 래치(latch)하여 자신의 캐쉬 메모리에 기록하는데, 이것은 레벨-2 캐쉬의 MLI 특성 때문이다.

(가) 블럭 K가 레벨-2 캐쉬에 VS, VM 상태로 존재하는 경우

레벨-2 캐쉬가 캐쉬 메모리를 엑세스하여 블럭 K를 전송하고, 상태는 바꾸지 않는다. 블럭을 전송받은 레벨-1 캐쉬는 VS 상태로 블럭을 기록한다.

(나) 블럭 K가 레벨-2 캐쉬에 DO 상태로 존재하는 경우

블럭 K의 마스터쉽(VM 상태)을 갖는 레벨-1 캐쉬가 전달한다. 버스상의 전송 동작을 감시하던 레벨-1 캐쉬의 스누퍼는 자신이 블럭 K에 대한 마스터쉽을 갖고 있으면, 캐쉬 메모리에서 블럭 K를 엑세스하여 데이타 버스로 전송하고, 자신의 상태를 VS로 바꾼다. 블럭을 전송받은 요청 레벨-1 캐쉬는 VS 상태로 블럭을 저장하고, 레벨-2 캐쉬 역시 블럭을 데이타 버스에서 래치하여 자신의 캐쉬 메모리에 VM 상태로 저장하여 그 블럭의 새로운 마스터(master)가 된다.

(다) 블럭 K가 레벨-2 캐쉬에 I 상태이거나 존재하지 않는 경우

(1) 지역 요청일 경우

S_R에 있는 블럭 K의 상태에 따라 전송 모듈이 결정된다. S_R은 블럭 K가 S 상태로 있으면 RC_SHD를 '1'로 구동하고, D 상태이면 RC_DIRTY를 '1'로 구동한다. 그리고 SBL 값도 전송해 준다.

① RC_SHD = '1' 일때, 블럭은 지역 메모리에서 전송된다. 블럭을 전송받은 요청 레벨-1 캐쉬와 레벨-2 캐쉬는 VS 상태로 블럭을 기록한다.

② RC_DIRTY = '1' 일때, 블럭의 전송은 레벨-2 캐쉬가 담당한다. 레벨-2 캐쉬가 레벨-SBL[K] 버스로 읽기 요청을 보내면 그 블럭의 마스터쉽을 갖는 캐쉬가 블럭을 전송해 준다. 블럭을 전송한 캐쉬와 블럭을 전송 받은 캐쉬의 상태는 VS가 된다. 레벨-2 캐쉬는 블럭 K를 요청 레벨-1 캐쉬로 전송해 준다.

③ RC_SHD = '0' & RC_DIRTY = '0' 일때, 블럭은 지역 메모리에서 전송된다. 블럭을 전송받은 레벨-1 캐쉬는 블럭 K의 마스터쉽을 갖게되어 VM 상태가 되고, 레벨-2 캐쉬는 DO 상태로 블럭 K를 기록하여 다른 클러스터의 요청시 블럭 K에 대한 엑세스 경로를 제공한다.

(2) 원격 요청일 경우

레벨-2 캐쉬가 레벨-B[K] 버스로 읽기 요청을 하여 블럭을 다른 클러스터로 부터 제공 받은뒤, 요청 레벨-1 캐쉬에게 공급한다. 레벨-2 캐쉬의 요청시 해당 S_R은 SBL[K]〈B[K]이면 SBL[K]를 B[K]로 바꾸고, 블럭의 상태가 D였으면 S 상태로 수정한다. 레벨-2 캐쉬가 상위 레벨 버스로 요청을 하였을때, 응답은 레벨-B[K] 버스의 상태 라인에 따라 다음과 같이 결정된다.

① SHD = '1'(레벨-B[K] 버스상에 블럭 K를 VS

상태로 소유한 캐쉬가 있는 경우)이면, 블럭을 소유한 레벨-2 캐쉬가 해당 블럭을 전송한다.

②SHD='0' & S_R에 S 상태(즉 SBL[K]>B[K])로 있으면, S_R은 SBL[K] 값을 요청 레벨-2 캐쉬에게 전달함으로써 요청 캐쉬가 다시 레벨-SBL[K] 버스로 요청하게 하여 ①처럼 블럭을 공급받게 한다.

③DIRTY='1'이면, 수정된 블럭을 소유한 레벨-2 캐쉬가 블럭을 공급하는데, 그 캐쉬의 상태가 VM이면 지연없이 즉시 전송을 하지만, 상태가 DO이면 수정된 블럭을 소유한 레벨-1 캐쉬에서 블럭을 가져와 공급한다. 블럭을 전송한 레벨-2 캐쉬와 블럭을 전송받은 레벨-2 캐쉬의 상태는 VS로 되고, S_R의 상태는 D에서 S로 바뀐다.

④DIRTY='0' & S_R 메모리에 D 상태(즉 SBL[K]>B[K])이면, S_R은 SBL[K] 값을 전달하여 요청 레벨-2 캐쉬가 레벨-SBL[K] 버스로 다시 요청하여 ③과 같이 블럭을 공급받게 한다.

⑤SHD='0' & DIRTY='0' & S_R에도 기록이 없으면, S_R은 자신의 지역 메모리에서 블럭을 가져와 요청 레벨-2 캐쉬로 전송하고, D 상태로 블럭을 기록한다.

①, ④와 같이 다른 클러스터의 캐쉬로 부터 블럭을 전송받은 경우에 레벨-1, 레벨-2 캐쉬의 상태는 모두 VS로 되며, ⑤와 같이 다른 클러스터 메모리로 부터 전송받은 경우에는 요청 레벨-1 캐쉬의 상태는 VM, 레벨-2 캐쉬의 상태는 DO로 된다.

4.3.4 기록 적중

프로세서가 기록을 요청한 블럭 K가 레벨-1 캐쉬에 VM이나 VS 상태로 있는 경우, 상태가 VM일 때는 즉시 기록이 수행되지만, VS 상태일 경우는 다른 공유 블럭을 모두 무효화 시킨 후 기록이 수행된다. 무효화될 블럭이 다른 클러스터에 공유되어 있을 경우는 레벨-2 캐쉬가 상위 레벨 버스로 무효화 방송을 하고, 무효화가 끝났음을 요청 레벨-1 캐쉬에게 알려준다. 무효화 방송이 끝나면 요청 레벨-1 캐쉬는 VM 상태로 되고, 해당 레벨-2 캐쉬는 DO 상태가 된다. 모든 기록 동작후에 마스터쉽은 기록을 수행한 레벨-1 캐쉬로 이동된다.

(가) 블럭 K가 레벨-2 캐쉬에 VS 상태로 존재

블럭 K가 지역 요청이고 S_R 메모리에 S 상태로 존재할때, S_R은 RC_SHD='1'과 SBL[K] 값을 레벨-2 캐쉬에게 보내고, 레벨-2 캐쉬는 레벨-SBL[K] 버스로 무효화 방송을 한다. 그러나 블럭 K가 지역 요청이고 RC_SHD='0'인 경우는 레벨-2 캐쉬는 상위 레벨 버스로 무효화 방송을 하지 않는다.

원격 요청일 경우는 레벨-2 캐쉬가 레벨-B[K] 버스로 블럭 K에 대한 무효화 방송을 한다. 그런데 해당 S_R 메모리에 SB[K]>B[K]이면(즉 레벨-SBL[K] 버스로 공유된 경우), S_R은 SBL[K] 값을 요청 레벨-2 캐쉬에게 알려 그 레벨-2 캐쉬가 다시 레벨-SBL[K] 버스로 무효화 방송을 하게 한다.

(나) 블럭 K가 레벨-2 캐쉬에 VM 상태로 존재하는 경우.

블럭이 클러스터 내에만 존재하므로 상위 레벨 버스로 무효화 방송을 할 필요없이, 레벨-2 캐쉬는 블럭의 상태를 DO로 변경한다.

4.3.5 기록 실패

기록 실패 동작은 읽기 실패후에 기록 적중 동작이 수행되는 것과 동일하다. 레벨-1 캐쉬는 버스로 기록을 위한 읽기(Read For Write) 요청을 보내어, 해당 블럭을 공유 메모리나 다른 캐쉬로 부터 전송 받는다.

4.3.6 블럭 대치

(가) 레벨-1 캐쉬 블럭 대치

프로세서가 요청한 블럭 K가 레벨-1 캐쉬 메모리에 없고, 대신 그 위치에 VM 상태를 갖는 다른 블럭이 존재할 때, 레벨-1 캐쉬는 기존의 블럭을 메모리로 기록-복귀 시킨뒤, 프로세서가 요청한 블럭을 가져와야 한다.

레벨-1 캐쉬가 블럭 K에 대한 기록-복귀를 요청할 때, 그 요청이 원격 요청이면 레벨-2 캐쉬가 레벨-B[K] 버스로 기록 복귀를 하고, 지역 요청인 경우는 지역 클러스터 메모리가 그 블럭을 대치하여 갱신하므로 레벨-2 캐쉬는 동작하지 않는다. 레벨-2 캐쉬는 블럭 K의 상태를 I로 바꾼다.

(나) 레벨-2 캐쉬 블럭 대치

레벨-2 캐쉬에서 대치할 블럭의 상태가 VS, VM일 때이다. 대치할 블럭의 상태가 VS인 경우는 레벨-1 버스로 무효화 방송을 하고, 동시에 레벨-B[K] 버스로도 블럭이 대치됨을 알리는 방송을 하여 다른 캐쉬들과 S_R이 알도록 한다. 블럭의 상태가 VM일 경우는 레벨-B[K] 버스로 기록-복귀 시키고, 레벨-1 캐쉬

에게 무효화 방송을 한다.

Ⅴ. 확장된 팬디드 프로토콜

계층 클러스터 구조에서는 빠른 전송 속도를 제공하는 팬디드 버스를 각 레벨 버스로 사용하고[14], 기존의 팬디드 버스를 계층 클러스터 구조에 적합하도록 확장한다. 그리고 확장된 팬디드 프로토콜을 사용하여 제안한 캐쉬 프로토콜에 대한 각 전송 형태를 규정하고, 시뮬레이션을 통한 시스템의 성능 평가를 수행한다.

5.1 데이타 전송 버스의 구조 및 전송 형태

5.1.1 확장된 데이타 전송 버스의 구조

팬디드 버스는 크게 데이타 전송 버스, 중재 버스, 인터럽트 버스, 그리고 유틸리티 버스로 구성된다. 이중 데이타 전송을 담당하는 버스는 데이타 전송버스로, 본 계층 클러스터 시스템에 사용될 확장된 데이타 전송 버스는〈그림 5-1〉과 같다. 확장된 부분은 상태 버스로 무효화를 위한 상태 라인과 S_R에서 구동하는 BLN, RC_DIRTY, RC_SHD 등이다.

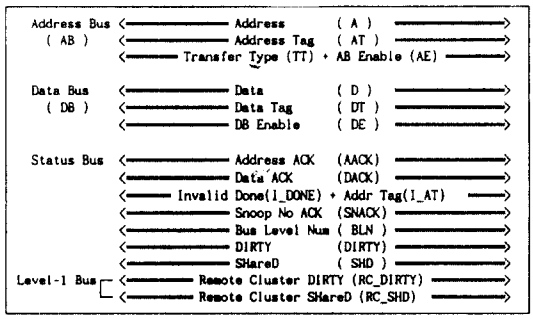


그림 5-1. 확장된 데이타 전송 버스의 구성.

5.1.2 기능 블럭

기능 블럭(function block)이란 데이타 전송 버스의 기능을 수행함에 있어서 기본적으로 필요한 버스 인터페이스 회로를 말한다. 확장된 팬디드 프로토콜은 캐쉬간의 직접 전송을 지원하기 위해 스누퍼에 응답 기능을 추가한다. 레벨-1 버스에는 〈그림 5-2〉의 (a)와 같이 버스 요청기(RQ), 메모리 응답기(RP), 스누퍼(SNP), DLC(Data Latch Controller) 4종류의 기능 블럭이 존재하고, 레벨-2 이상의 버스에는 〈그림 5-2〉의 (b)와 같이 버스 요청기, 스누퍼, S_R_DLCi, 3 종류의 기능 블럭이 존재한다. 레벨-1 버스에서 데이타 전송은 RQ와 RP 혹은 RQ와 스누퍼간에 이루어지고, 레벨-2 이상의 버스에서는 RQ와 스누퍼 사이에서 이루어진다. CC_DLC는 P_RQ로 전달되는 데이타를 래치하기 위해 사용되고, S_R_DLCi는 레벨-i 버스상에서 자신의 메모리 영역에 대한 전송시 데이타를 래치하기 위해 사용된다.

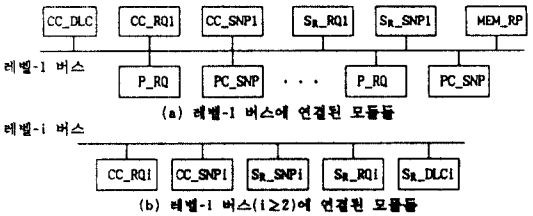


그림 5-2. 각 레벨 버스의 기능 블럭.

5.1.3 데이타 전송 버스의 기본 주기

확장된 팬디드 프로토콜에서 기본 주기는 어드레스 기본 주기(Address Cycle : AC), 데이타 기본 주기(Data Cycle : DC), 그리고 어드레스 데이타 기본주기(Address and Data Cycle : A&DC), 무효화 기본 주기(Inv Cycle : IC)등 4가지의 기본 주기가 있다. 이중에서 무효화 기본 주기는 레벨-2 캐쉬의 스누퍼가 레벨-1 캐쉬의 무효화 요청이 끝났음을 알리기 위해 I_DONE 신호와 요청 레벨-1 캐쉬의 주소(I_AT)를 전송할 때 사용하는 주기로, 중재를 거치지 않고 레벨-2 스누퍼가 구동할 수 있다. 아래 〈표 5-1〉과 〈그림 5-3〉은 본 데이타 전송버스에서 규정한 기본 주기의 특성을 요약한 것이다.

5.2 전송 형태에 따른 버스 동작

5.2.1 Read For Read(RFR)

캐쉬 일관성 프로토콜의 읽기 실패에 해당되는 전송 형태로 모든 레벨 버스에 존재한다. RFR의 버스상의 동작 순서는 〈그림 5-4〉와 같다.

5.2.2 Read For Write(RFW)

프로세서의 쓰기 요청이 레벨-1 캐쉬에서 실패했을 때, 레벨-1 캐쉬는 P_RQ로 RFW을 요청한다. RFW의 버스 동작과 버스상에 뜨는 신호는 RFR의 경우와 일치하지만, 캐쉬의 내부 동작은 RFR와 다르다.

표 5-1. 데이타 전송 버스의 기본 주기들.

Name	# of Phase	From	To	Function
어드레스 기본 주기	3	RQ	RP, SNP	Addr Bus 정보 전송
데이타 기본 주기	1	RP, SNP	RQ, SNP	Data Bus 정보 전송
어드레스데이타기본주기	4	RQ	RP, SNP	A&D Bus 정보 전송
무효화 기본 주기	1	CC SNP1	P_RQ	I_DONE, I_AT 전송

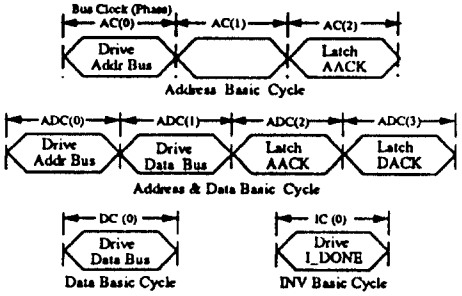


그림 5-3. 기본 주기의 버스 동작.

5.2.3 INValid(INV)

레벨 1 캐쉬에 VS 상태로 있는 데이타를 변경하고자 할때 PC_CNTR는 P_CNTR는 P_RQ로 INV를 요청한다. INV 전송은 각 레벨 버스에 존재하며 동작 순서는 <그림 5-5>와 같다.

5.2.4 WRite Back(WRB)

WRB은 레벨-1 캐쉬에서 대치되어 나가는 블럭의 상태가 VM일때 메모리로 기록 복귀, S_R에 D 상태로 있는 블럭의 읽기 요청시에 S_R_RQ1이 해당 블럭을

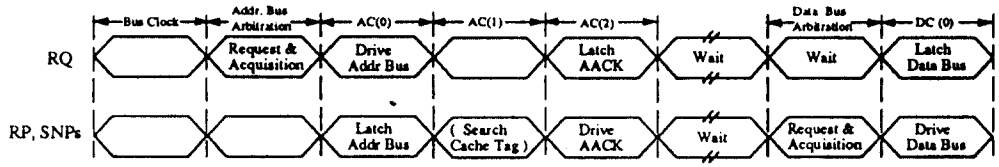


그림 5-4. RFR 동작 순서.

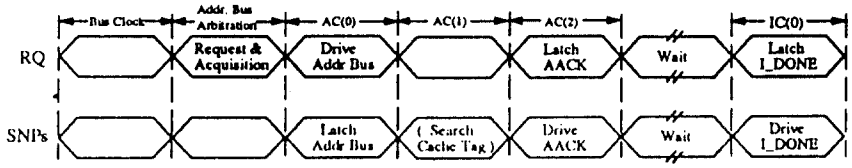


그림 5-5. INV 동작 순서.

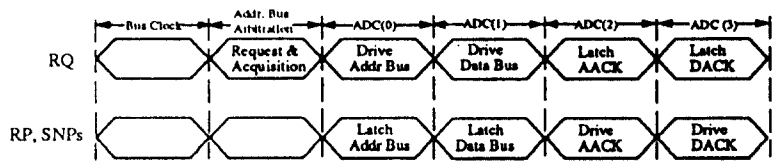


그림 5-6. WRB 동작 순서.

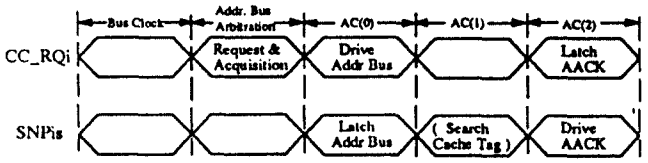


그림 5-7. SBR 동작 순서.

메모리로 기록 복귀하는 경우, 레벨-2 캐쉬에 VM 상태의 블럭을 대치하는 경우의 기록 복귀, 3가지로 구분된다. WRB의 동작 순서는 〈그림 5-6〉과 같다.

5.2.5 Shared Block Replacement(SBR)

SBR는 레벨-2 캐쉬에서 VS 상태의 블럭을 대치할 때 발생하는 형태로 레벨-2 이상의 버스에만 존재하며 동작 순서는 〈그림 5-7〉과 같다.

5.3 시뮬레이션

5.3.1 시뮬레이션 성능 평가 요소

시뮬레이션을 하기 위한 가정은 다음과 같다. 먼저 프로세서가 외부 메모리를 요청할 확률은 지수 확률분포(exponential distribution)를 따르며, 메모리 모듈을 접근할 확률은 국부성(locality)과 균일 확률 분포(uniform distribution)를 따른다고 가정하자. 블럭의 상태는 블럭의 공유율, 레벨-1 캐쉬 적중률, 레벨-2 캐쉬 적중률에 따라 결정된다.

시뮬레이션 모델에서 계층 클러스터 구조의 시스템 성능을 평가하기 위한 파라미터(parameter)들은 다음과 같다.

(1) 클러스터의 갯수
(2) 클러스터당 프로세서의 갯수
(3) 클러스터당 메모리 모듈의 갯수
(4) On-chip 캐쉬 적중률
(5) 레벨-1 캐쉬 적중률
(6) 레벨-2 캐쉬 적중률
(7) 읽기 대 쓰기 비율
(8) 지역 요청 및 원격 요청의 비율
(9) 블럭의 공유율
(10) 메모리 큐의 갯수
(11) WRB의 확률

위와 같은 파라미터 값의 변화를 시뮬레이션 모델에 적용하여 측정하는 버스의 성능 측정치는 어드레스 버스 이용률, 데이타 버스 이용률, 평균 메모리 엑세스 시간 등이다.

5.3.2 시뮬레이션 결과

(가) 레벨-2 캐쉬 적중률의 변화에 따른 결과

256개의 프로세서, 128개의 메모리 블럭을 갖는 3-레벨 계층 클러스터 구조에서 레벨-1 캐쉬의 적중률을 0.95라 하고 레벨-2 캐쉬 적중률을 변화시켜갈때, 어드레스 버스의 사용률은 〈그림 5-8〉와 같다. 레벨-1 버스 사용률은 레벨-2 캐쉬 적중률과 관계없이 거의 균일하게 유지됨을 볼 수 있고, 레벨-2, 3 버스는 레벨-2 캐쉬 적중률이 증가할수록 크게 감소함을 볼 수 있다. 이것은 레벨-2 캐쉬 적중률이 높을수록 대부분의 요청이 클러스터내에서 적중되어 처리되기 때문에 상위 레벨 버스로 나가는 요청이 줄어듬에 기인한다.

(나) 블럭 공유율의 변화에 따른 결과

〈그림 5-9〉는 블럭의 공유율에 5%에서 50%까지 변화시킬때, 버스 사용률을 측정한 결과이다. 그림은 레벨-1 버스를 통하여 메모리를 요청할 확률(LR), 레벨-2 버스를 통하여 메모리를 요청할 확률(RR1), 레벨-3 버스를 통하여 메모리를 요청할 확률(RR2)을 6:3:1로 준 경우와 4:3:3으로 준 경우를 비교한 것이다. 레벨-3 버스의 사용율을 보면 6:3:1인 경우는 공유율이 30% 미만일때 적정수준의 버스 사용률(30%)

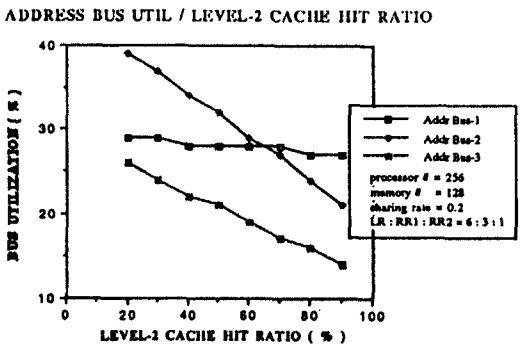


그림 5-8. 레벨-2 캐쉬 적중률에 따른 어드레스 버스의 사용률.

을 보이고, 4:3:3인 경우는 공유율이 5%일때부터 30% 이상의 버스 사용률을 보인다. 따라서 본 계층 클러스터 구조에서는 국부성을 보장한 작업 할당이 이루어질때 우수한 성능을 낼 수 있으리라 기대된다.

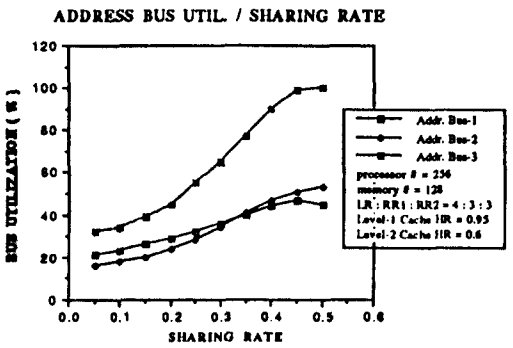


그림 5-9. 공유율의 변화에 따른 어드레스 버스의 사용률.

VI. 결 론

본 논문에서는 Wilson의 계층 캐쉬/버스 구조와 클러스터 구조의 단점을 보완한 확장 가능한 계층 클러스터 구조를 제안하였고, 이 구조하에서의 효율적인 캐쉬 일관성 프로토콜을 설계하였다. 그리고 캐쉬 일관성 동작을 팬디드 프로토콜로 구현하여 버스상의 동작을 설명하고, 시스템의 성능 평가를 위해 시뮬레이션을 수행하였다.

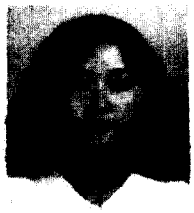
계층 클러스터 구조는 클러스터 메모리를 사용하기 때문에 작업의 국부성을 증가시킬 수 있고 외부로의 통신 트래픽을 줄일 수 있다. 또한 하위 레벨 캐쉬의 내용을 포함하는 클러스터 캐쉬를 사용함으로써 캐쉬 일관성 간섭으로부터 하위 레벨 캐쉬들을 보호하여 성능 향상을 도모한다. 또한 각 클러스터들은 한개 이상의 상위 레벨 버스에 연결되기 때문에 클러스터간의 통신을 신속하게 할 수 있다. 제안한 캐쉬 일관성 프로토콜은 기록-복귀와 기록-무효화 방법을 사용하고 블럭의 마스터쉽 개념을 채택하여 무효화 방송을 줄이도록 설계되었다. 또한 캐쉬간의 직접 전송을 지원하고 팬디드 프로토콜을 사용함으로써 시스템의 성능 향상을 도모한다.

256개의 프로세서로 구성된 3-레벨 계층 클러스터 구조에 대한 시뮬레이션을 수행한 결과, 시스템의 성능에 영향을 미치는 요인은 작업의 국부성, 레벨-1, 2 캐쉬의 적중률, 블럭의 공유율로 밝혀졌다. 분석 결과, 국부성이 보장되는 작업 할당이 이루어지거나, 레벨-1, 2 캐쉬의 적중률이 비교적 높거나, 또는 블럭의 공유율이 25% 미만이면, 각 레벨 버스의 사용률은 적정 수준(30%미만)을 나타내 전체 시스템 성능이 향상되었다. 따라서 본 논문에서 제안한 계층 클러스터 구조는 중대형 다중프로세서 시스템에 적합한 구조로 기대된다.

참 고 문 헌

1. J.K. Archibald and J.L. Baer, "Cache Coherence Protocols : Evaluation Using a Multiprocessor Simulation Model," ACM Trans. on Comp. Systems, Vol. 4, pp. 273-298, Nov. 1986.
2. J.K. Archibald, "A Cache Coherence Approach For Large Multiprocessor System," ACM Proc. Supercomputing, pp. 337-345, 1988.
3. S. M. Mahmud, "A Distributed Cache Coherence Protocol for a Hierarchical Bus Multiprocessor System," IEEE Proc. of 3rd. Annual Paral. Processing Symp., pp. 103-117, 1989.
4. L. Rudolph and Z. Segall, "Dynamic Decentrallized Cache Schemes for MIMD Processors," IEEE 11th Int. Symp. on Comp. Arch., pp. 340-347, 1984.
5. M. S. Papamarcos and J. H. Patel, "A Low-overhead Coherence Solution for Multiprocessors with Private Cache Memories," IEEE 11th Int. Symp. on Comp. Arch., pp. 348-354, 1984.
6. A. W. Wilson Jr., "Hierarchical cache/bus architecture for shared memory multiprocessors," Proc. Int. Symp. on Comp. Arch., pp. 244-252, 1987.
7. J. R. Goodman and P. J. Woest, "The Wisconsin Multicube : A New Large Scale Cache Coherent Multiprocessor," IEEE 15th Int. Symp. on Comp. Arch., pp. 422-431, 1988.
8. P. Lenoski and J. Laudon, "The Directory-based Cache Coherence Protocol for the DASH Multiprocessor," IEEE 17th Int. Symp. on Comp. Arch., pp. 148-159, May 1990.
9. S. Haridi and E. Hagersten, "The Cache Coher-

ence Protocol of the Data Diffusion Machine," Proc. PARLE'89 vol. 1, Springer-Verlag, pp. 1-18, 1989.
10. Q. Yang, G. Thangadurai, and L. N. Bhuyan, "Design of an Adaptive Cache Coherence Protocol for Large Scale Multiprocessors," IEEE Trans. on Paral. and Dist. Systems, vol. 3., pp. 281-293, May 1992.
11. L.N. Bhuyan and A. K. Nanda, "Multistage Bus Network(MBN) : An Interconnection Network Cache Coherent Multiprocessors," Proc. of the 3rd IEEE Symp. on Paral. and Dist. Processing, pp. 780-787, 1991.
12. A. L. Decegama, The Technology of Parallel Processing, Englewood Cliffs, NJ : Prentice-Hall, pp. 101-103, 1989.
13. J.L. Baer and W.H. wang, "On the Inclusion Properties for Multilevel Cache Hierarchies," IEEE 15th Int. Symp. on Comp. Arch., pp. 73-80, 1988.
14. 최창열, 박병관, 박승규, 오길록, "공유 메모리와 단일 버스로 구성되는 다중프로세서의 하드웨어 성능 분석," 한국 정보과학회 논문지, Vol. 16, No. 5, pp. 309-408, Sep. 1989.



朴 信 民(Sin Min Park) 정회원
1991년 2월 : 서강대학교 전자계산학과 졸업
1993년 2월 : 서강대학교 대학원 전자계산학과 졸업(공학석사)
1993년~현재 : 한국통신 통신망연구소 전임연구원
※주관심분야 : Computer Architecture, Computer Network

崔 昌 勳(Chang Hoon Choi) 정회원
1988년 2월 : 명지대학교 전자계산학과 졸업
1990년 2월 : 서강대학교 대학원 전자계산학과(공학석사) 졸업
1990년 1월~9월 : 대우통신 기술개발부 근무
1992년~현재 : 서강대학교 대학원 전자계산학과 박사과정 재학중
※주관심분야 : Computer Architecture, parallel processing system

金 聖 天(Sung Chun Kim)정회원
1975년 : 서울대학교 공과대학 공업교육학 (전기전공)학사
1976년~1977년 : 동아컴퓨터(주) Sys. Eng.
1977년~1978년 : 스페리유니맥 Sales Rep.
1979년 : Wayne State Univ. 컴퓨터공학 석사
1982년 : Wayne State Univ. 컴퓨터공학 박사
1982년~1984년 : 캘리포니아주립대 조교수
1984년~1985년 : 금성반도체(주) 책임연구원
1986년~1989년 : 서강대학교 공과대학 전자계산소 부소장
1989년~1991년 : 서강대학교 공과대학 전자계산학과 학과장
1985년~현재 : 서강대학교 공과대학 전자계산학과 조교수 (1985. 8~1987. 8), 부교수(1987. 9~1992. 8), 교수(1992. 9~현재)
1989년~현재 : 한국정보과학회 병렬처리시스템 연구회 부위원장, 대한전자공학회 및 한국통신학회 논문지 편집위원(1991, 1993), 한국정보과학회 학회지 부위원장(1993)
※주관심분야 : 병렬처리시스템(Parallel Computer Architecture, Interconnection Network), Neural Network, Computer Network