

# 다중 프로세서 시스템에서의 공유 메모리 최적화 연구

김중수\*, 문중욱\*, 임강빈\*, 정기현\*, 최경희\*\*

\* 아주대학교 전자공학부, \*\* 아주대학교 정보통신 전문 대학원

E-Mail : promise@csl.ajou.ac.kr

## A Study on Shared Memory Optimization for Multi-Processor System

Jong Su Kim\*, Jong Wook Moon\*, Kang Bin Yim\*, Gihyun Jung\*, Kyunghye Choi\*\*

\*School of Electronics Engineering Ajou University,

\*\*The Professional Graduate School for Info. & Comm. Ajou University

### 요 약

고속 I/O 를 갖는 Loosely coupled 다중 프로세서 시스템은 데이터의 처리 성능 향상과 I/O 집중화에 따른 병목 현상을 줄여줄 수 있다. 이 때 프로세서간의 데이터 전송에 사용되는 공유메모리는 그 구성과 이용 방법에 따라 시스템의 성능에 많은 영향을 미친다. 본 연구에서는 공유메모리의 사용 방법을 비동기, 메일박스를 통한 인터럽트 전달 인지 방식으로, I/O 사용방법을 고속 이더넷(IEEE 802.3u)으로 한 시스템 모델에서 다중 프로세서 시스템 구성에 필요한 공유메모리의 최적 사용량을 입출력 데이터의 Bandwidth 와 Burstness 관점에서 실험을 통해 분석하였다.

### 1. 서론

마이크로 프로세서 및 주변 장치들의 지속적인 성능향상에도 불구하고, 정보 처리율을 더욱 높이기 위해 다중 프로세서 시스템을 이용하는 연구가 많이 진행되고 있다. 특히 근래 인터넷의 폭발적 사용으로 인해 이에 연결할 수 있는 이더넷 제어 장치가 많이 사용되고 있는데, 이 고속 I/O 를 통해 들어오는 데이터들은 빠른 속도와 많은 양 때문에 이들을 처리해 줄 고성능 다중 프로세서 시스템이 필요해 지는 경우가 있게 된다. 다중 프로세서 시스템을 통해 입력 데이터의 분산 처리가 가능해지면 계산 능력과 I/O 사용 능력을 향상시킬 수 있게 된다. 이러한 시스템에서 프로세서간 통신을 위한 방식은 성능과 가격면에서 큰 영향을 미치게 된다.

본 연구의 2 장에서는 다중 프로세서 시스템 구성에 대한 중요 설계 항목에 대해 설명하고, 3 장에서는 결과 예측을 위한 간략한 모델링과 제한 조건에 대해 설명한다. 4 장에서는 이에 대한 실제 모델을 통한 실험과 이의 결과를 분석하고, 5 장에서 전체적인 결론과 향후 연구 방향에 대해 서술하도록 한다.

### 2. 관련 연구

지금까지 처리 속도를 높일 수 있는 많은 방법이

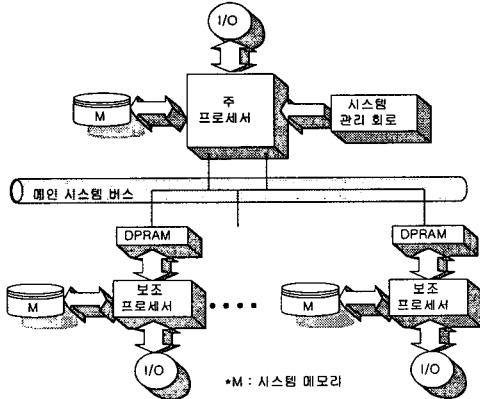
구상되었으며, 그 중의 한 방편으로 다중 프로세서 시스템을 사용하는 방법이 있다. 다중 프로세서 시스템은 하나의 프로세서가 모든 일을 처리하던 방식에서 벗어나 여러 프로세서가 작업을 나눠 처리할 수 있도록 하여, 폭주를 방지하고 동시 작업을 수행할 수 있도록 지원 한다. 특히 특수 기능에서 고 효율성이 필요한 시스템에서는 특정 시스템 자원 사용 빈도가 지극히 높게 되는데, 다중 프로세서 시스템은 이의 집중화에 따른 병목현상을 방지할 대책이 되기도 한다.[1]

이런 다중 프로세서 시스템에서는 여러 프로세서간의 유기적인 결합을 위한 정보 공유가 필수가 되며, 이 정보 공유를 위한 통신 자원의 성능이 시스템에 큰 영향을 미치는 변수가 된다.[3] 본 연구에서는 이 자원 공유를 위해 사용되는 매개체 중 많은 데이터 양과 속도를 감안하여 공유메모리(FIFO, Dual Port RAM 등)에 초점을 둔다. 이 공유메모리는 정보 전달 속도에 있어서는 고성능을 발휘하나 용량당 단가가 비싸다는 단점이 있어 불필요한 낭비를 줄일 수 있다면 시스템의 성능과 경제성에 큰 효과가 있게 된다. 본 논문에서는 공유메모리 사용량을 늘려도 더 이상 성능 향상이 일어나지 않는 성능 대 가격의 최적 지점을 실험을 통해 구명함으로써 공유메모리 사용량을 최적화할 수 있도록 한다.

2.1 시스템 구성 모델

프로세서의 행동을 정의할 프로그램 영역과 실행 작업에 필요한 데이터 영역을 위해서 각 프로세서는 메모리 자원을 필요로 하며, 이의 구성 위치는 시스템의 특성을 결정짓는다.[3] 예를 들어 프로세서간의 자원 공유와 유기적 결합력이 큰 시스템의 경우 시스템 메모리를 공유하는 방식이 많이 사용되며, 공유 메모리 사용에 대한 문제는 이 메모리의 접근 방식에서 나타난다. 한편 프로세서들의 독립적 연산 처리가 가능한 경우 시스템 메모리는 분리되어 구성될 수 있으며, 공유 메모리는 이들간의 정보 전달을 위해 사용된다.[1]

본 논문에서 연구 모델로 후자를 선택하였으며, 그림 1과 같이 자체적으로 시스템 자원(메모리, I/O 등)을 갖는 프로세서 시스템들을 1대 1 방식의 공유메모리(Dual-Port RAM)로 엮어 전체를 구성하며, 시스템 버스를 통한 메모리의 Burst (DMA) 접근, Single Beat 접근이 가능하도록 한다.



< 그림 1. 다중 프로세서 시스템 구성 >

2.2 통신 모델

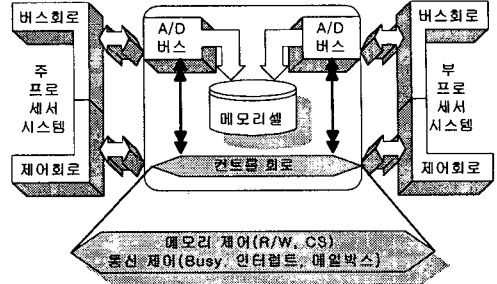
통신 자원의 최소화나 통신 지연 부하를 줄이기 위해 다중 프로세서 시스템의 통신 방식 설계 시 행하는 고려사항 중 전송 모델(병렬, 순차), 전송 타입(동기, 비동기), 전송 프로토콜(FIFO, Multi-Port Memory, Bus block, Bus non-block) 등은 시스템의 성격을 규정짓는 주요한 인자들이다.[3]

본 연구에서는 정보 처리 속도에 초점을 맞춰 데이터의 병렬 전송을 선택하고, 이기종 프로세서간의 연산 및 입출력 처리 속도 차이를 보상하기 위해 전송 방식을 비동기로 채택하였다. 또한 데이터 흐름이 양방향일 수 있도록 하며, 데이터 전송의 신뢰성을 위해 통신 제어 방법으로 메일박스 방식을 사용한다.

이 과정에서는 데이터 겹쳐 쓰기 등의 오류를 방지하고 상대방에 데이터 전달을 인지시키기 위해 데이터를 전송하는 측에서 데이터를 써넣고 이의 포인터 정보를 메일박스에 써 넣으면 데이터를 수신하는 측에 인터럽트가 발생되어 공유메모리 상에 변화가 있

음을 알려 주도록 한다.[7]

특히 Dual-Port 메모리를 사용하는 전략에 있어서 통신 포트를 통해 들어오는 데이터의 가장 큰 크기의 메모리 버퍼를 N 개 구성한 후 이의 포인터와 사이즈 정보를 이용하여 관리하는 방법을 사용한다.[8]



< 그림 2. Dual-port RAM 을 이용한 통신설계 >

3. 분석 모델

현재 모델링 한 Loosely coupled 다중 프로세서 시스템을 살펴보면 주 프로세서는 고속 I/O 데이터를 입력 받아 이를 전처리 한 후, 해당 부 프로세서로 전달하고 이를 처리한 부 프로세서는 처리결과를 출력하게 되는 I/O 처리 중심의 시스템이다.

3.1 공유메모리 최적화 전제 조건

공유메모리 사용량 계산을 위한 전제 조건은 다음과 같다.

시스템의 성능이 더 이상 증가하지 않는 최대까지 용량을 산출하고, 데이터 손실을 방지하기 위해 입력 Bandwidth 는 출력 Bandwidth 보다 작도록 한다.

시스템의 I/O Burstness<sup>2</sup> Rate 는 단위시간당 동일 bandwidth 내에서 그 Burst 한 구간의 길이 비율로 정한다.

3.2 시스템 타이밍 분석 모델

주, 부 프로세서 시스템에서 소요 시간 분석을 위한 그림은 아래와 같으며, 데이터의 흐름은 그림의 번호 순이다.

1. I/O 입력 단에서는 데이터 손실 방지를 위해 I/O 장치에 연결된 내부 FIFO 를 사용하며, 이에 따른 접근 지연에 대해서는 고려하지 않는다.[5]
2. 고용량 고속 I/O 에서의 데이터를 CPU 가 실시간 연산 처리 할 수 없는 경우, 이의 손실을 방지하기 위해 시스템 메모리(SDRAM 등)에 DMA 를 이용하여 데이터를 이동시켜야 한다.
3. CPU 는 시스템 메모리에 넣어진 데이터를 읽어

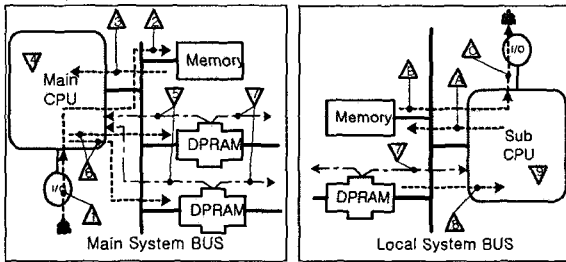
많이 사용하는 폴링, 인터럽트 I/O 방식 중 상대적으로 구현이 어려우나 성능면에서 뛰어난 인터럽트 I/O 방식을 사용한다.[4]

<sup>2</sup> I/O 데이터들 사이의 간격이 규격상 최소간격으로 유지되는 것을 뜻함

<sup>1</sup> 현재 메시지 전달방식에서 수신 스케줄링 정책으로

수행하고자 하는 특정 전 처리 작업을 하는데 이를 위해 주 메모리로부터의 데이터 패치 작업이 필요하다.

4. CPU 내부에서 입력된 데이터를 처리하기 위한 작업을 수행한다.
5. 동기화 과정에서는 변화된 데이터의 위치를 알려 주기 위해 정보를 써 넣고, 인터럽트를 인지한 후 이를 읽어 들이며, 메일 박스 정보 분석을 통해 데이터 오류방지를 위한 공유 메모리 접근 지연 작업 등을 수행한다.
6. 공유 메모리에 접근하여 정보를 써 넣는 것은 시스템 버스를 통하여 일반 메모리에 접근하는 것과 같으며, 메모리의 특성과 접근 시간에 의존한다.



< 그림 3. 주, 부 시스템 접근 순서 및 모델 >

이에 따라 주 시스템에서 I/O로부터 들어온 데이터가 처리되어 상대방측에 전달되는데 걸리는 시간은 이들 과정에서의 시간 합으로 나타낼 수 있다.

다른 인자의 경우 시스템이 설계되면 동일 비율의 값을 갖으나, 공유메모리 접근 지연과 메일 박스 정보 분석 시간 값은 시스템의 I/O, 처리속도, 시스템 메모리, 공유 메모리의 크기에 큰 영향을 받게 되는데 다중 처리 시스템의 성능 향상을 위해서는 이 지연 시간을 줄여야 한다. 이 지연 시간을 보상하기 위해 현 시스템에서는 다른 인자에 대한 영향은 고정시키고 공유 메모리의 크기에만 영향이 미치도록 하였다.

### 3.3 부 시스템 타이밍 분석 모델

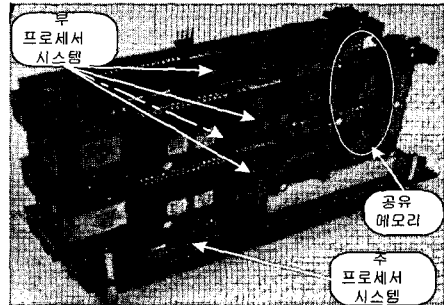
부 시스템에서도 역시 같은 방식의 데이터 접근이 이루어지므로 해당 항목의 시간 값 계산은 같은 방식으로 이루어진다. 다만 시스템의 버스 속도 등의 실제 값만이 다른 내용을 갖게 된다.

## 4. 실험 모델

고속 이더넷( Fast Ethernet ) 인터페이스는 근래 IT 산업의 증가 추세에 따라 내장형 시스템에도 많이 적용되는 I/O 중의 하나로서 많은 부하(예, 고용량의 빈번한 사용으로 인한 집중화 현상 등)를 가지며, 이용되는 시스템의 특성 상 많은 경우 이의 데이터 처리 속도 또한 보장 받기를 원한다(QoS 등). 이에 다중 프로세서 시스템에서는 네트워크 데이터에 대한 분산 작업 처리를 실험하여 시스템 성능과 데이터 그리고 공유 메모리량의 관계를 확인한다.

### 4.1 실험 환경

그림 4의 실험 모델은 이더넷 컨트롤러가 내장된 내장형 프로세서를 이용하여 시스템 버스 접근 부하를 줄이고, 고속 메모리인 SDRAM 과 Dual-Port Memory 를 사용하여 구성한 시스템이다. 주 프로세서는 통신 기능 안정성과 성능을 보장하는 모토로라사의 PowerPC 코어 기반의 MPC860T 이고, 부 프로세서는 ARM7TDMI 코어 기반의 삼성 NetARM S3C4530 이다. 고속 이더넷 연결은 MII 인터페이스를 사용한다.



< 그림 4. 실험 모델 >

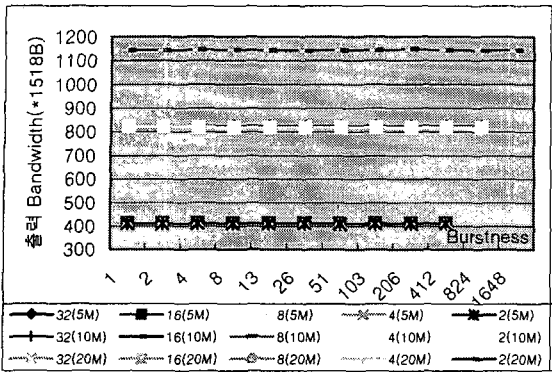
실제 네트워크는 부하 모델이 변하여 시스템의 성능 실험에 부적합하므로 네트워크 패킷 생성기(Packet Generator)를 이용하여 부하 모델을 같게 설정한다. 이더넷(IEEE 802.3) 상에서는 패킷의 크기가 64~1518Byte로 가변적인데 본 실험에서는 최소, 최대 크기의 임의의 UDP 패킷을 생성하여 실험하며, Bandwidth 는 5, 10, 20Mbps 를 이용한다. 부 프로세서 시스템에서는 입력 데이터에 대해 많은 계산을 수행하도록 하여 네트워크를 통한 데이터 출력 Bandwidth 가 약 14Mbps 가 되도록 한다.

Burstness란 입력 패킷간의 군집성을 의미한다. 실험에서의 Burstness 값은 고속 이더넷에서 규정한 패킷간 시간 간격(Inter-Packet Gap)이 최소(0.96us)인 패킷들의 개수를 의미하는 것으로 같은 Bandwidth 에서의 상대적 수치이다. 같은 Bandwidth 를 가지나 버퍼의 크기에 따라 패킷 손실이나 지연을 가져올 수 있게 된다.

### 4.2 실험 결과

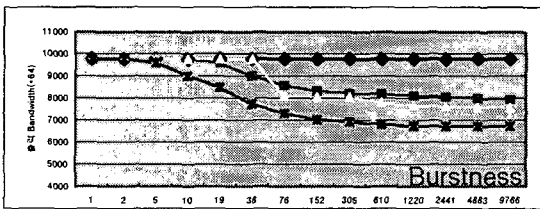
본 연구에서는 최적화 전제 조건에 의해 손실 없는 시스템에 대해 논의 하고 있는데 입력이 20Mbps 이고 출력이 14Mbps 로 제한된 상황이라면 이의 지속에 따라 데이터 손실이 발생한다. 그림 5 와 그림 6-(c)에서는 출력 Bandwidth 가 버퍼나 Burstness 에 상관없이 한계 출력 Bandwidth 에 의해 제한 받아 일정하게 손실됨을 증명하고 있다.

또한 입력 Bandwidth 가 출력 Bandwidth 보다 작은 경우인 5, 10Mbps 의 경우에도 순간 손실율을 줄이기 위해 SDRAM 에서의 입력 버퍼링을 충분히 한 설정으로 인해 Burstness 나 DPRAM 크기 변화에 상관없이 1518Byte 크기의 패킷은 100% 처리가 가능함을 알 수 있다.

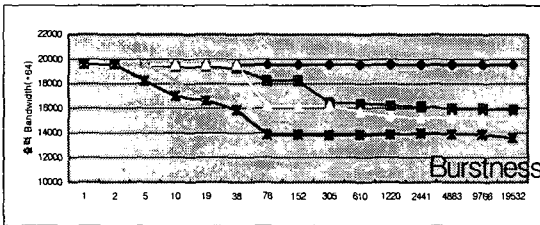


< 그림 5. 1518-Byte 입력 패킷에 대한 처리량 >

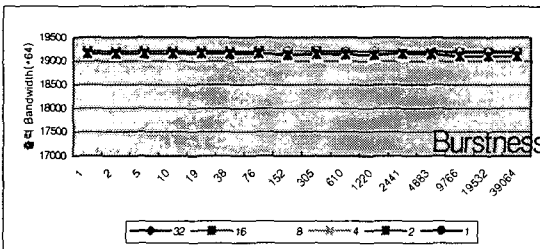
한편 64Byte 크기를 가지는 패킷은 1518-Byte 와 같은 Bandwidth 를 가지기 위해 20 배 이상의 패킷 개수를 필요로 하는데, 이는 공유메모리 상에 데이터 갱신 횟수가 잦아 이에 따른 동기화 작업에 많은 시간이 소요됨을 뜻한다.



(a) 5Mbps 입력 Bandwidth 에서의 영향



(b) 10Mbps 입력 Bandwidth 에서의 영향



(c) 20Mbps 입력 Bandwidth 에서의 영향

< 그림 6. 64-Byte 입력 패킷에 대한 처리량 >

그림 6 의 (a)와 (b)는 5M, 10Mbps 의 Bandwidth 에서 공유 메모리 크기를 변화 시켜가며 Burstness 에 대한 처리 가능 여부를 실험한 값이다. 입력 Bandwidth 가 출력 Bandwidth 보다 낮은 경우 일단 데이터 처리 속도가 충분함은 같은 Bandwidth 를 가지고 계산된 1518-

Byte 의 예에 의해 증명되므로 동기화 작업 시간 과다 가 시스템에 데이터 손실을 가져 올 수 있음을 의미 한다. (높은 Burstness 에 대해 적은 공유메모리 크기를 갖는 항목에서의 데이터 손실 발생 )

이 데이터 손실을 없애기 위해 중간 버퍼링 역을 하는 공유메모리의 크기를 늘려 동기화 지연 시간을 최대한 낮춰야 한다. 그림 6-(a),(b)에서는 데이터 집중 비율을 의미하는 Burstness 에 대해 공유메모리 크기가 증가되면 데이터 손실이 줄 수 있음을 보여주고 있는데, 모든 Burstness 정도에서도 손실이 없도록 하기 위한 공유메모리 크기는 입력 Bandwidth 가 5M, 10Mbps 인 경우 16 < 패킷 버퍼 < 32 에서 찾을 수 있다.

한편 일정 Burstness 에 대해서는 공유 메모리 크기가 증가해도 시스템의 성능에 더 이상 영향을 주지 못하는 지점을 발견할 수 있다. 예로 입력 5Mbps, 최대 Burstness 가 16 인 환경에서는 패킷 버퍼 16 이상은 무의미하다는 것을 알 수 있으며, 이 지점이 시스템 구성의 최적화 지점이 된다.

### 5. 결론 및 향후 연구 과제

본 연구에서는 고속 데이터를 입출력으로 갖는 다중 프로세서 시스템에서 프로세서간의 공유메모리 크기의 최적화에 대한 연구를 하였다. 최적화되는 공유 메모리의 크기는 프로세서 처리 속도, 데이터 버퍼링 크기, 입력 데이터의 Burstness 와 Bandwidth 와 밀접한 관계를 가지고 있으며, 이의 상관 관계 규명으로 예측 가능한 최적화 설계가 가능함을 네트워크 데이터의 특성에 따른 공유 메모리 사용 실험으로 보였다.

현재는 버퍼 크기에 미치는 요인 중 Burstness 와 Bandwidth 에 대한 변수만을 고려하였으나, 향후 다른 변수들에 대한 종합적인 영향을 실험하고 모델과의 비교 검증을 통하여 이론적용이 가능하도록 할 것이다.

### 참고문헌

- [1] Mano, M.Morris. "Computer System Architecture", 3rd Ed. Prentice-Hall, 1993
- [2] Sundar Iyer, "Analysis of a Memory Architecture for Fast Packet Buffers", 2001 IEEE Workshop on High Performance Switching and Routing, May 2001
- [3] G. Gogniat, M. Auguin, "Communication synthesis and HW/SW integration for Embedded System Design", CODES/CASHE'98, 1998
- [4] Juan Carlos Gomez, "The CLAM Approach to Multithreaded Communication on Shared-Memory Multiprocessor : Design and Experiments", IEEE Transactions on Parallel and Distributed Systems, Volume 9, pp.36-49, January 1998
- [5] Motorola, "860T Design Advisory 0.3", [www.mot-sps.com](http://www.mot-sps.com)
- [6] Cypress, "Understanding Asynchronous Dual-Port RAMs", [www.cypress.com/whitepaper/](http://www.cypress.com/whitepaper/), 1997
- [7] Raoul A.F.Bhoedjang "User level network interface protocols", IEEE Computer, pp.53-61, November 1998
- [8] Dave Dunning, "The Virtual Interface", IEEE Micro, pp.66-76, March/April 1998