

2013 SoC Conference

2013 SoC 학술대회

- 경북대학교 글로벌플라자 효석홀
- 2013년 5월 2일(목)-5월 3일(금)

■ 주 최 대한전자공학회 SoC 설계 연구회

■ 주 관 경북대학교 / ETRI SW-SoC 융합 R&BD센터

■ 후 원 Cadence Korea / 리버트론 / ELECS / 동운아나텍 / 아이앤씨테크놀로지 /
휴인스 / 한국전자통신연구원(ETRI) / 대구컨벤션뷰로 / IDEC 플랫폼센터



Session D1 : Digital

세션번호	발표시간	발표장소	좌장
D1	09:30 - 11:00	206호	이용환 교수 (금오공대)

■ D1-1

대용량 메인 메모리 시스템에서 DIMM 트리의 최적 차수에 대한 연구

장영종, 김영규, 문병인
(경북대학교)

■ D1-2

패턴영상의 Edge 강화 2차원 필터의 하드웨어 구현

최대성, 서인석, 조호상, 강봉순
(동아대학교)

■ D1-3

SoC 설계 플랫폼을 위한 DSP core 포팅 및 검증

왕지범, 박주성
(부산대학교)

■ D1-4

Design of an Easily Retargetable Halt-Mode On-Chip Debugger for RISC Processors

Jingzhe Xu, Seungpyo Jung, Youngju Park, Sangman Kim, Jusung Park
(부산대학교)

■ D1-5

Fast human detection method using cumulative temporal data

Sanghun Kim, Dong-Gon Yoo, Gyujin Bae, Young Hwan Kim
(포항공과대학교)

■ D1-6

메모리 효율을 높인 SURF 알고리즘 기반 특징점 추출 하드웨어 가속기 설계

정창민, 박종일, 곽재창, 이광엽
(서경대학교, (주)씨엔에스테크놀로지)

2013년 SoC 학술대회

대용량 메인 메모리 시스템에서 DIMM 트리의 최적 차수에 대한 연구

장영종¹, 김영규², 문병인^{1,*}¹경북대학교 IT대학 전자공학부, ²경북대학교 전자전기컴퓨터학부

전화: (053)950-7580, E-mail: bihmoon@knu.ac.kr

A Study on the Optimal Degree of the DIMM Tree in the Large Capacity Main Memory System

Young-Jong Jang¹, Young-Kyu Kim², Byungin Moon^{1,*}¹School of Electronics Engineering, Kyungpook National University,²School of Electrical Engineering and Computer Science, Kyungpook National University

요 약

전통적인 컴퓨터 시스템에 주로 사용 되었던 멀티드롭(multi-drop) 기반의 메모리 접근 방식은 신호의 무결성(signal integrity) 문제로 인해 대용량 메인 메모리 시스템의 구현에는 적합하지 않다. 이를 해결하기 위해 DIMM 트리 기반의 메모리 접근 방식들이 제안 되었지만 DIMM 트리의 효율적인 구조에 대한 구체적인 연구는 많이 부족한 실정이다. 이에 본 논문은 DIMM 트리의 트리 차수에 따른 DIMM 트리 성능에 대한 연구를 수행하였다. 시뮬레이션을 위해 16 TB의 메인 메모리를 가지는 DIMM 트리를 모델링 하여 트리의 차수를 가변 할 수 있게 하였으며, Zipf 분포도를 응용하여 10억 개의 주소로 구성된 workload를 생성하여 시뮬레이션 하였다. 그리고 시뮬레이션 결과를 통해 하드웨어 자원 대비 성능 측면에서 최적의 차수를 가지는 DIMM 트리를 제안하였다.

Abstract

The multi-drop based memory access method is mainly used in the traditional computer systems. However, it is not appropriate for implementation of large capacity main memory systems because of signal integrity issues. In order to resolve this problem, DIMM tree based memory access method was suggested but, specific studies on the efficient structure of the DIMM tree is lacking. Thus, this paper examines studies on performance of DIMM tree according to tree degree of DIMM tree. For the simulation, we modeled the DIMM tree that have a main memory of 16 TB. It can switch degree of tree and then applies Zipf distribution probability to generate workload with address of a billion. Through the simulation results, we propose DIMM tree that has a optimal degree in the aspect of performance compared with hardware architecture.

Keywords : DIMM Tree, Many-DIMM Tree, Main Memory Database,

I. 서 론

대용량 메인 메모리 시스템은 느린 응답 속도의 디스크 대신 메인 메모리에 모든 데이터를 상주시킴으로써

시스템의 성능을 극단적으로 향상시킨 시스템이다. 최근에는 DRAM의 가격 하락 및 빅데이터 처리 등과 같은 사회적 기조로 인해 대용량 메인 메모리 시스템에 대한 중요성이 고조되고 있다^[1]. 하지만 전통적인 컴퓨터 시스템에서 주로 사용하던 멀티드롭(multi-drop)

* 교신저자

기반의 메인 메모리 접근 방식들은 신호 무결성(signal integrity) 문제로 인해 메모리 버스 당 연결 가능한 노드의 개수가 제한적이게 되므로 대용량 메인 메모리 시스템에 적용하기에는 적합하지 않다^[2]. 이에 대한 대안으로 DIMM(dual in-line memory module) 트리 기반의 시스템 구조가 제안되었지만 지금까지의 관련 연구들은 DIMM 트리의 실효성을 증명하는 수준에서 크게 벗어나지 못하였으며 DIMM 트리의 구체적인 구조들에 대한 연구는 부족한 실정이다^[3]. 이에 본 논문은 대용량 메인 메모리 구성을 위한 DIMM 트리의 구조에 대한 연구로서 최적 트리 차수에 관한 연구를 진행하였으며, 자체적인 모델링 및 시뮬레이션을 통해 최적의 차수를 가지는 DIMM 트리 구조를 제안한다.

II. 본 론

멀티드롭 기반의 메인 메모리 접근 방식은 신호의 무결성 문제로 인해 연결할 수 있는 DIMM의 개수가 제약 받으며 버스의 클럭 주파수가 높을수록 연결 가능한 DIMM의 개수는 더욱 감소한다^[2]. 그러므로 많은 수의 DIMM을 사용하는 대용량 메인 메모리 시스템에서 멀티드롭 기반의 메인 메모리 접근 방식을 사용하는 것은 효율적이지 않다. 이러한 문제점을 극복하기 위해 제안되었던 DIMM 트리는 기존의 멀티드롭 기반의 메모리 접근 방식에 포인트 투 포인트 링크(Point-to-Point Link) 방식을 접목시켜 대용량의 메인 메모리 시스템 구현을 가능하게 한다^[3].

DIMM 트리는 그림 1과 같이 2개 이상의 DIMM들을 트리 구조로 연결한 것으로서, 연결되는 DIMM의 개수가 증가하더라도 트리의 레벨(level)만 늘어날 뿐 메모리 버스 당 연결되는 DIMM의 개수는 변하지 않는 구조이다. 이러한 DIMM 트리 구조에서는 트리의 각 레벨에 따라 메모리 접근을 위한 지연시간(latency)이 각각 달라진다. 즉 레벨1에 속하는 DIMM에 접근할 때는 한 홉(hop)의 지연시간이 필요하지만 레벨2에 위치한 DIMM에 접근할 경우에는 두 홉의 지연시간이 필요하게 된다^[4]. 이러한 점을 개선하기 위해 DIMM 트리는 그림 1과 같이 한 홉의 지연시간으로 접근 가능한 레벨의 DIMM들을 fast partition으로 정하고, 나머지 DIMM들은 slow partition으로 구분하였다. Slow partition은 기존의 메인 메모리와 기능이 크게 다르지 않으며 fast partition은 빈번하게 참조되는 데이터들을 위한 캐시로 사용되어 트리 구조에서 발생하는 성능 저

하를 최소화 하였다^[3].

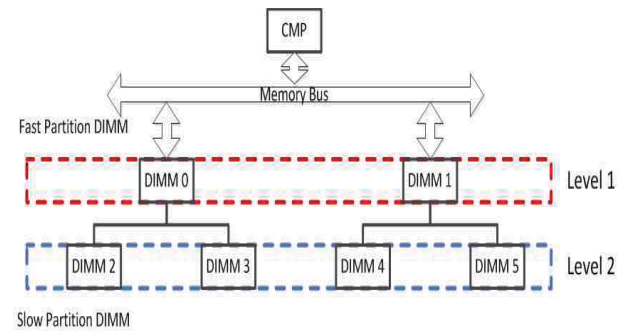


그림 1. 차수가 2인 DIMM 트리 구조의 Fast/Slow Partition
 Fig. 1. Fast/Slow partition of The DIMM Tree by Degree 2

DIMM 트리는 트리의 차수(degree)에 의해 구조가 결정되는데 트리의 차수는 메모리 버스에 연결되는 DIMM의 개수에 의해 결정된다. 기존 연구들에서는 DDR3-1600 DRAM으로 DIMM 트리 구조에 대한 실험을 진행하였기 때문에 이론적으로 구성 가능한 최대 트리 차수를 4로 고정하고 실험하였다^[4]. 하지만 DIMM 트리 구조에서는 차수의 값에 따라 fast partition과 slow partition의 트리 구조와 정책들이 달라지기 때문에 신호 무결성 문제를 배제하더라도 DIMM 트리의 효율적인 구조에 대한 연구가 필요하다. 특히, DIMM 트리에서 차수는 fast partition의 크기를 결정하며, 이는 전통적인 컴퓨터 시스템의 cache set associative mapping과 동작 방식은 유사하지만 캐시로 사용되는 fast partition의 메모리 크기가 물리적으로 증가하는 차이점이 있기 때문에 하드웨어 자원 대비 성능에 대한 검증이 반드시 필요하다. 이에 본 논문에서는 DIMM 트리의 하드웨어 구조를 모델링하고 동일한 workload에 대해 다양한 차수의 DIMM 트리 구조를 하드웨어 자원 대비 성능 측면에서 시뮬레이션 함으로써 최적의 DIMM 트리 차수를 구하는 연구를 진행하였다.

III. 실험 방법

본 논문에서 모델링한 DIMM 트리 구조의 하드웨어적 환경은 표 1과 같다. 모델링 된 시스템의 메인 메모리 크기는 총 16 TB로 설정하였으며 이를 위해 4 GB DIMM 4096개를 DIMM 트리의 slow partition으로 구성하였다. 그리고 fast partition의 크기는 4 GB

DIMM으로 DIMM 트리의 차수에 따라 8 GB에서 128 GB까지 메모리의 용량을 증가시켰다. 또한, 16 TB 크기의 메인 메모리를 고려하여 44비트 주소를 사용하였으며, 실제 시스템과 유사한 특성의 데이터 지역성 (data locality)을 가진 workload를 얻기 위해 Zipf 분포^[5]를 통하여 주소를 생성하고, 이를 편집하여 10억 개의 주소로 구성된 workload를 만들었다.

표 1. 모델링 환경
Table1. Environment for Modeling

DIMM 트리 차수	2, 4, 8, 16, 32
Fast Partition 메모리 크기	8 GB ~ 128 GB (DIMM 트리 차수 × 4 GB DIMM)
Slow Partition 메모리 크기	16 TB (4 GB DIMM × 4096 개)
페이지 크기	4 KB
Address Width	44 bits address
Workload 개수	10억 개

DIMM 트리의 DIMM들 역시 페이지 단위로 관리되기 때문에 각각의 DIMM마다 1 MB 크기의 페이지 테이블을 구현하여 모델링에 적용하였으며 실험의 간소화를 위해 TLB(translation look a side buffer)는 생략하였다.

44비트 주소 중 하위 12비트는 페이지 오프셋으로 사용하였으며 그 다음 20비트는 페이지 인덱스로 사용하였다. 그리고 상위 12비트는 캐시 동작을 위한 태그로 사용하였다. 트리의 차수에 따른 fast partition의 DIMM들을 구별하기 위한 별도의 주소 필드가 필요하지만 모델링의 간소화를 위해 페이지 오프셋의 상위 비트를 이용하여 구현하였다. 모델링에 사용된 address scheme에 대한 필드들은 그림 2와 같다.



그림 2. DIMM 트리를 위한 어드레스 맵핑 스킴
Fig. 2. Address Mapping Scheme for DIMM Tree

DIMM 트리의 차수에 따른 성능 변화를 알아보기 위해 본 논문에서는 10억 개의 workload가 처리되는 동안 fast partition에서 발생한 page faults를 카운트하

여 성능 지표로 사용하였다.

IV. 실험 결과

본 논문에서는 DIMM 트리 차수 별 page faults를 측정하기 위해 각 차수에 대하여 동일한 workload를 입력하여 시뮬레이션을 수행하였다. 그림 3은 본 논문의 시뮬레이션 결과로서, DIMM 트리의 차수를 2, 4, 8, 16, 32로 증가시킴에 따른 page faults 발생 비율을 측정한 것이다. 이 중 트리의 차수가 8이하인 경우에는 차수가 증가함에 따라 page faults 발생 비율이 33% 감소하였고, 트리의 차수가 8에서 16으로 증가하였을 경우에는 page faults 발생 비율이 17% 감소하였다. 하지만 트리의 차수가 16이상인 경우에는 page faults 발생 비율이 0.2% 감소하여 매우 미미한 수준이었다. 이러한 결과는 본 논문에서 사용한 workload가 가지는 주소 분포에서 트리의 차수가 16인 fast partition의 메모리 크기만으로도 데이터의 데이터 지역성을 충분히 반영 할 수 있는 것으로 분석된다. 그러므로 본 논문에서 모델링한 시스템 환경에서는 트리의 차수가 16인 DIMM 트리 구조가 메모리 자원 대비 성능 측면에서 최적인 것을 알 수 있다.

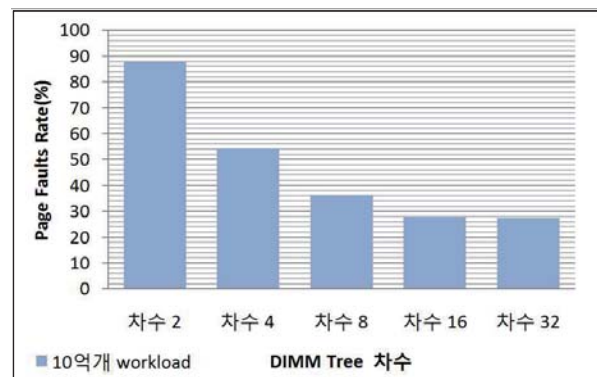


그림 3. DIMM 트리의 차수에 따른 page faults rate
Fig. 3. Page Faults Rate by DIMM Tree's Degree

V. 결론

본 논문은 대용량 메인 메모리 시스템에서 대용량의 메인 메모리를 구성하기 위한 DIMM 트리의 구조에 대한 연구로서, 메모리 자원 대비 최적의 성능을 낼 수 있는 DIMM 트리의 차수에 대한 연구를 수행 하였다. 시뮬레이션 결과 트리의 차수가 16인 경우에 메모리 자원 대비 성능이 최적인 것으로 나타났으며, 이러한

연구 결과는 대용량 메인 메모리 시스템이 요구되는 시스템을 설계 시 메인 메모리의 크기 및 메모리 버스 구조를 결정할 때 참고할 수 있기 때문에 메인 메모리 시스템 분야의 발전에 크게 기여할 것으로 기대 된다. 단, 이러한 결과는 시스템의 메인 메모리 크기 및 workload의 주소 분포 등에도 영향을 받으므로 이 부분에 대한 추가적인 연구가 필요하다. 또한, DIMM 트리의 구조 및 노드 관리 정책 등에 대한 연구들도 함께 진행할 계획이다.

감 사 의 글

본 논문은 경북대-삼성전자 반도체 산학협력위원회 연구과제에 의해 지원된 연구 결과입니다.

참 고 문 헌

- [1] H. Plattner and A. Zeier, "In-memory data management: an inflection point for enterprise applications," Springer, pp. 11-23, 2011.
- [2] B. Jacob, S. Ng, and D. Wang, "Memory Systems: cache, dram, disk," Morgan kaufmann, pp. 377-379, 2008.
- [3] K. Therdsteerasukdi, G. S. Byun, J. Ir, G. Reinman, J. Cong and M. F. Chang, "The DIMM tree architecture: a high bandwidth and scalable memory system," in Proc. of IEEE 29th International Conf. on Computer Design, pp. 388-395, University of massachusetts, USA, October 2011.
- [4] K. Therdsteerasukdi, G. S. Byun, J. Ir, G. Reinman, J. Cong and M. F. Chang, "Utilizing radio-frequency interconnect for a many-DIMM DRAM system," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, Vol. 2, no. 2, pp. 210-227, June 2012.
- [5] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in Proc. of IEEE Conf. on Computer and Communications Societies, pp. 126-134, New York, USA, March 1999.