

저전력 고성능 뱅크-승격 TLB 구조

(A High Performance and Low Power Banked-Promotion TLB Structure)

이 정 훈 [†] 김 신 덕 ^{**}

(Jung-Hoon Lee) (Shin-Dug Kim)

요 약 TLB(translation lookaside buffer) 성능 향상을 위한 전형적인 방법으로 TLB로 하여금 더 많은 엔트리 수를 지원하는 방법, 페이지 크기를 증대시키는 방법, 그리고 다중 페이지 크기를 지원하는 방법이다. 현재 TLB 성능을 높이기 위한 가장 좋은 방법으로 다중 페이지 크기를 지원하는 연구가 효과적이다. 다중 페이지 크기를 지원하는 방법들 중에서 가장 적합한 것은 운영체제나 컴파일러로부터 일정한 정보를 받아서 가장 적합한 페이지 크기를 TLB에 할당하는 것이다. 그러나 운영체제의 시스템 영역에서는 이러한 방식이 가능하나 사용자 영역에서는 현실적으로 이러한 방식을 지원하기 어렵기 때문에 우리는 사용자 영역까지 운영체제의 지원 없이 이중 페이지를 지원할 수 있는 새로운 TLB 구조를 제안하고자 한다. 고성능을 보장하기 위한 방법으로 이중페이지를 하드웨어 방식으로 지원하는 방식을 제시하고, 또한 저전력의 효과를 높이기 위하여 하나의 완전 연관 TLB를 두 개의 서버 TLB로 나누는 방식을 사용하였다. 이러한 두 가지 방식을 내장형 프로세서에 적합하도록 하나로 집적화 하여 저전력과 고성능의 효과를 모두 얻을 수 있었다. 시뮬레이션 결과에 따르면 적은 엔트리만을 사용하고자도 높은 성능 향상효과를 얻을 수 있으며 또한 완전 연관 TLB에 비해 거의 50%이상 소비전력을 줄이는 효과를 얻을 수 있다.

키워드 : 메모리 계층구조, 메모리 관리, 변환참조버퍼(TLB), 성능 평가, 저전력 시뮬레이션

Abstract There are many methods for improving TLB (translation lookaside buffer) performance, such as increasing the number of entry in TLB, supporting large page or multiple page sizes. The best way is to support multiple page sizes, but any operating system doesn't support multiple page sizes in user mode. So, we propose the new structure of TLB supporting two pages to obtain the effect of multiple page sizes with high performance and at low cost without operating system support. we propose a new TLB structure supporting two page sizes dynamically and selectively for high performance and low cost design without any operating system support. For high performance, a promotion-TLB is designed by supporting two page sizes. Also in order to attain low power consumption, a banked-TLB is constructed by dividing one fully associative TLB space into two sub-fully associative TLBs. These two banked-TLB structures are integrated into a banked-promotion TLB as a low power and high performance TLB structure for embedded processors. According to the results of comparison and analysis, a similar performance can be achieved by using fewer TLB entries and also power consumption can be reduced by around 50% comparing with the fully associative TLB.

Key words : memory hierarchy, memory management, translation lookaside buffer, performance evaluation, and low power simulation.

· 이 연구는 삼성 MMU 프로젝트에 의해 수행된 과제입니다.

† 학생회원 : 연세대학교 컴퓨터학과
ljh@yonsei.ac.kr

** 종신회원 : 연세대학교 컴퓨터학과
sdkim@yonsei.ac.kr

논문접수 : 2001년 9월 4일
심사완료 : 2002년 1월 16일

1. 서론

정보화 사회가 급속히 세분화되고 거대화됨에 따라 멀티미디어 및 통신 응용분야를 중심으로 다양한 형태의 내장형 시스템이 보급되고 있다. 특히 이동 통신 단말기, MP3 재생기, 포터블 컴퓨터, PDA (personal

digital assistance)와 같은 포터블 시스템의 출현은 범용의 마이크로 프로세서 대신 고성능 내장형 프로세서에 대한 요구를 증대시키고 있다. 오늘날 컴퓨터 시스템 내의 다양한 분야 중에서 TLB (translation look-aside buffer)는 메모리 접근 지연시간 (memory access latency)과 소비 전력을 줄이고 전체 시스템의 성능 향상을 높이기 위한 기본적이고 효과적인 방법중의 하나로써 제안되어왔다. TLB는 가상 주소 (virtual addresses)를 물리 주소 (physical addresses)로 변환하기 위한 페이지 테이블 (page table)을 구성하는 캐쉬 메모리이다 [1]. 페이지화 된 가상 메모리 (paged virtual memory)를 지원하는 대부분의 컴퓨터에서 TLB는 주소 변환에 소요되는 평균 시간 (average address translation time)을 줄이기 위해서 사용되어진다.

TLB 성능향상을 위한 전형적인 방법은 크게 세 가지 방식으로 구분되어진다. 첫째, TLB로 하여금 더 많은 엔트리 수 (entry number)를 지원하는 방법, 둘째 페이지 크기 (page size)를 증대시키는 방법, 셋째, 다중 페이지 크기 (multiple page size)를 지원하는 방법이다 [2, 3]. 그러나 TLB 엔트리 수가 증대되어 진다면, 메모리 참조의 지연 (latency)이라는 역효과가 나타나며, 또한 일반적으로 TLB는 CAM (content addressable memory)으로 구현되어지기 때문에 참조 때마다 매번 많은 엔트리를 비교해야 함으로 전력 면에서도 상당히 불리한 요소가 많다. 또한 페이지 크기를 증대시킬 경우, 메모리의 사상 (mapping)의 적용 정도 (coverage)가 증가한다는 큰 장점을 가지게 되지만, 페이지 내부 단편화 (internal fragmentation)의 증가로 메모리 낭비가 심해지고 사상되는 페이지의 수를 제한함으로 프로세서의 수가 제한을 받게 된다는 단점도 가지게 된다. 그러므로 현재 TLB 성능을 높이기 위한 가장 좋은 방법으로 다중 페이지 크기를 지원하는 연구가 효과적이다.

다중 페이지 크기를 지원하는 방법들 중에서 가장 적합한 것은 운영체제나 컴파일러로부터 일정한 정보를 받아서 가장 적합한 페이지 크기를 TLB에 할당하는 것이다. 그러나 운영체제의 시스템 (kernel) 영역에서는 이러한 방식이 가능하나 사용자 (user) 영역에서는 현실적으로 이러한 방식을 지원하기 어렵기 때문에 우리는 사용자 영역까지 운영체제의 지원 없이 이중 페이지를 지원할 수 있는 새로운 TLB 구조를 제안하고자 한다. 이중 페이지의 가장 큰 장점은 내부 단편화를 줄일 수 있으면서도 필요에 따라 큰 페이지의 사용으로 높은 성능 향상을 기대할 수 있다. 그러나 기존의 이중 페이지

지원 방식은 운영 체제의 도움을 받아서 수행되기 때문에 하드웨어적으로 표준화된 구조를 제시하기가 어렵고 운영체제의 수정과 자원이 필요하였기 때문에 제한된 응용분야에 한정된 TLB구조로 사용되어왔다.

일반적으로 캐쉬나 TLB와 같은 프로세서의 내부 메모리에서 소모되는 전력은 전체 칩에서 소모되는 전력의 상당부분을 차지한다. 예를 들어, 최근의 대표적인 RISC 내장형 프로세서인 StrongARM-110 [4]의 전력 소모를 살펴보면, 명령어 캐쉬, 데이터 캐쉬, TLB의 전력 소모가 칩 전체의 전력 소비의 27%, 17%, 17%를 차지하는 것을 알 수 있다. 이렇게 내부 메모리가 많은 전력 소모를 보이는 이유는 다음과 같다. 첫째, 이러한 온 칩 메모리 시스템을 구성하는 태그와 데이터 배열들은 프로세서의 빠른 클럭 주파수를 지원하기 위하여 주로 전력 소모가 많은 정적 메모리 (static RAM)로 구현된다. 특히, 완전 연관 (fully-associative) 방식의 TLB를 구현하는데 사용되는 CAM은 내부 비교 로직과 부가적인 매치 라인 (match line)들로 인해 SRAM보다도 훨씬 많은 전력을 소비한다. 둘째, 이러한 온 칩 메모리 시스템은 매우 자주 접근되는 경향이 있다. 특히, TLB는 프로그램 수행동안 매 클럭 사이클마다 접근되어야 한다. 셋째, 이러한 메모리 시스템 접근 시에 발생할 수 있는 접근 실패 (miss)는 또 다른 대용량의 온 칩 메모리 시스템을 접근하거나 오프 칩 메모리 접근을 위해서 I/O 패드를 구동해야 한다. 일반적으로 I/O 패드의 정전용량 (capacitance)은 온 칩 정전 용량 보다 훨씬 크다. 따라서 온 칩 메모리 접근 실패 횟수를 줄이는 것이 저전력 메모리 시스템을 설계하기 위한 기본 접근 방법이 된다. 이처럼 TLB가 캐쉬와 더불어 전력 소모가 많은 블록임에도 불구하고 지금까지 저전력 메모리 시스템에 대한 연구는 주로 캐쉬 메모리에 초점을 맞추고 진행되어 왔다 [5, 6, 7, 8]. 일반적으로 저전력 시스템 설계를 위한 고려는 설계 과정의 각 단계에서 이루어 질 수 있으며 이는 상위 레벨의 알고리즘 선택, 시스템 집적, 아키텍처 설계에서부터 하위 레벨의 게이트/회로 설계, 공정 단계를 포함한다. 예를 들어 저전력 메모리 시스템의 설계는 낮은 공급 전압을 사용하거나 [9], 공정 기술의 향상, 저 전력 메모리 셀의 설계 [10], 그리고 메모리 시스템의 구조적인 향상 [5, 6, 7, 8] 등의 방법에 의해 달성될 수 있다. 이러한 다양한 저전력 설계 단계 중 아키텍처, 알고리즘 및 시스템 레벨에서의 저전력 설계 방식은 공정 기술의 변화나 회로/로직 설계를 통한 방식보다 적은 연구 노력과 설계 비용으로 큰 효과를 도출시킬 수 있으며, 저전력 설계 중 가장 포

팔적인 개념을 다루기 때문에 가장 중요한 비중을 차지한다. 따라서 본 연구는 상위 레벨의 설계 단계에서 고성능/저전력 내장형 프로세서를 위해 TLB에 요구되는 특성을 고려하고, 성능, 전력 소모, 설계비용을 총체적으로 고려한 실험과 분석을 통해 고성능/저전력 TLB 구조를 제안하고자 한다.

시뮬레이션 결과에 따르면 전력과 성능을 모두 고려한 에너지*지연시간 곱의 성능 향상율은 완전 연관 TLB에 비해 약 99.8%, micro-TLB에 비해 약 19.2%, banked-TLB에 비해 약 24.2%, 그리고 victim-TLB에 비해서는 약 24.4%의 성능 향상을 얻을 수 있었다.

이 논문의 나머지 부분은 다음과 같다. 관련 연구는 제 2장에서 소개되어지며, 제 3장은 제안된 TLB의 구조와 동작원리에 대한 기술을 설명한다. 제 4장에서는 성능 평가 지표와 성능 비교 그리고 소비 전력에 대한 시뮬레이션 결과를 비교·분석한다. 제 5장에서 결론을 맺는다.

2 관련 연구

오늘날 고성능 마이크로 또는 내장형 프로세서들은 명령어 TLB와 데이터 TLB를 내장하고 있으며 대부분 완전 연관 TLB 구조를 사용하고 있다. 완전 연관 구조를 이용하여 TLB를 구성할 경우 작은 TLB 크기로 높은 성능 향상을 기대할 수 있지만 참조 시간이 길어지고 높은 전력을 소비하는 단점을 가지기도 한다. 그러나 TLB 접근 실패 시 처리해야 하는 지연시간은 대단히 높음으로 고성능을 보장하는 완전 연관 구조가 적합하다고 할 수 있다. 표 1은 완전 연관 TLB 구조를 사용하고 있는 프로세서의 예이다.

표 1 완전 연관 TLB 구조를 사용하고 있는 상용화 프로세서

프로세서	명령어 TLB 엔트리 수	데이터 TLB 엔트리 수
MIPS R10000	8-entry	64-entry
Alpha21164	48-entry	64-entry
UltraSPARC-II	64-entry	64-entry
PowerPC620	64-entry	64-entry
StrongARM	32-entry	32-entry

성능 향상을 높이기 위한 방법으로 운영체제의 지원을 받지 않고 이중 페이지를 지원하는 TLB로는 완전 서버블록 (complete-subblock) TLB가 있다 [11]. 이

방식은 낮은 가격 대 성능 비를 고려한 것으로 TLB 하나의 엔트리에 태그 부분은 16Kbyte의 페이지를 나타내는 하나의 가상 페이지 번호와 4개의 물리적인 페이지 번호로 구성되어 진다. 이는 기본 페이지 크기를 4KB로 두고 16KB의 효과를 내기 위한 방법이며, 또한 운영체제의 지원을 받지 않는 구조임으로 4개의 4KB의 물리적인 페이지 번호를 저장하고 있어야 한다. 만약 가상 페이지 번호가 순차적으로 접근되어 진다면 좋은 성능을 보일 것으로 기대되지만 항상 엔트리에 물리적인 페이지 번호를 저장할 공간을 마련해 두고 있기 때문에 비순차적인 경우에는 엔트리의 공간 낭비가 우려된다. 또한 하나의 엔트리가 대체 (replacement) 되는 경우 4개의 4KB 물리적인 페이지 번호 정보가 모두 소실됨으로 성능 저하를 초래하는 가장 큰 단점으로 지적될 수 있다. 그러나 제안된 TLB 구조는 완전 서버 블록 TLB의 단점을 보완하면서 메모리 공간의 사용을 더욱 효과적으로 이용할 수 있는 구조라 할 수 있다.

저전력의 효과를 얻기 위한 TLB 구조로는 계층적인 마이크로 TLB (filter-TLB) 구조 [12]가 일반적으로 이용되고 있다. 마이크로 TLB 구조는 Main TLB 보다 상위 계층에 작은 크기의 TLB를 운영하는 방법으로, 소비전력의 측면에서는 효과적인 구조이지만 데이터 TLB의 경우 낮은 성능으로 일반적으로 명령어 TLB에 국한되어 사용되고 있다. 예로 SH4 와 PA-RISC2.0은 4개의 마이크로 명령어 TLB를 단일 TLB (Unified-TLB) 상위 계층에 위치시켜 명령어에 대한 사이클 내에 참조가 일어나고, 참조 실패인 경우 다음 사이클동안 단일 TLB를 참조하는 메커니즘을 사용하고 있다. 이외에도 저전력 TLB를 위해 CAM 메모리 셀 자체를 변형시키는 방법 [10], 낮은 공급 전압을 이용하는 방법 [9], 그리고 뱅크 구조를 이용한 방법들이 있다 [13]. 메모리 셀 자체를 변화시키는 것은 하드웨어 비용이 증가하는 단점을 가지게 되고 또한 낮은 공급 전압을 제공하는 방법은 다른 기술적인 문제를 해결해야 하는 어려운 작업이라 할 수 있다. 또한 순수한 뱅크 구조는 저전력 TLB에는 효과적이지만 하나의 뱅크에 편중될 확률이 높음으로 다른 뱅크의 활용도의 저하로 성능을 감소시키는 단점들이 있다. 이에 제안된 TLB 시스템은 순수한 뱅크 구조의 단점을 극복하는 방법을 제시하고 저전력의 효과를 기대할 수 있는 방향을 제시하고자 한다.

3 제안된 TLB의 구조적 특징과 동작 원리

이 장에서는 제안된 TLB의 동작 모델과 구조적 특징

을 설명하고 새로운 TLB 시스템을 제안하게 된 배경과 동기에 대해서 상세히 살펴 볼 것이다.

3.1 제안된 TLB 시스템 구조

이 연구의 주목적은 간단하지만 저 전력과 높은 성능 향상을 보이는 TLB 시스템을 설계하는 것이다. 따라서 주어진 TLB 공간을 최적화 시키는 방법으로 이중 페이지를 효과적으로 이용할 수 있는 구조와 저전력을 고려한 뱅크 구조로 구성된 TLB 시스템을 제안하고자 한다. 제안된 banked-promotion TLB (BP-TLB) 구조는 크게 작은 페이지 크기 (예로 4KB, 8KB, 또는 16KB)를 지원하는 기본적인 완전 연관 TLB (fully-associative TLB)와 큰 페이지 크기 (예로 16KB, 32KB, 또는 64KB)를 지원하는 완전 연관 TLB의 두 부분으로 구성되어 있다. 또한 저전력의 효과를 높이기 위하여 뱅크 구조를 이용하였다. 일반적으로 작은 페이지는 가상 페이지를 지원하는 기본 페이지이다. 이후부터 작은 페이지 크기를 지원하는 TLB를 banked-TLB, 큰 페이지 크기를 지원하는 TLB를 promotion-TLB라 명명한다.

Banked-TLB 의 각 엔트리는 하나의 가상 페이지 번호 (virtual page number: VPN)와 하나의 물리적인 페이지 번호 (physical page number: PPN)로 구성되며 promotion-TLB는 하나의 VPN와 네 개의 PPNs로 구성된다. 이처럼 물리적 페이지 번호 네 개를 하나의 엔트리에 저장함으로써 운영체제의 지원을 완전히 배제 하면서 이중 페이지를 하드웨어적으로 지원하기 위한 구조라 할 수 있다. promotion-TLB 하나의 엔트리는 banked-TLB 여러 개의 연속적인 엔트리 사상 정보에 해당하며 반드시 한번 이상 생성된 주소에 대해서만 큰 페이지 TLB로 들어올 수 있다. 제시하는 BP-TLB를 구성하는 한 가지 설계의 예로 작은 페이지 크기가 4KB이고 큰 페이지 크기가 16KB인 경우를 가정할 때 promotion-TLB의 운용 방식은 다음과 같다. Banked-TLB내에 한 개의 큰 페이지에 해당하는 4개의 순차적인 작은 페이지 번호 중 3개의 순차적인 가상 페이지가 이미 존재할 경우, 다음에 참조하는 작은 페이지가 해당하는 큰 페이지의 4번째 가상 페이지이고 작은 페이지 TLB에서 접근 실패가 발생하는 경우를 고려하자. 이 경우 4개의 순차적인 작은 페이지는 한 개의 큰 페이지로 재구성 가능하며 이를 페이지 승격 (promotion)이라 정의한다. 이때 MMU는 네 번째 작은 페이지에 대한 접근 실패를 처리하게 되고 그 처리 동안 작은 페이지 TLB내의 3개의 순차적 가상 페이지 번호와 접근 실패 처리중인 작은 가상 페이지 번호를 큰

페이지 TLB내에 한 개의 엔트리로 등록하게 된다. 이렇게 한 개의 큰 페이지 엔트리로 묶인 네 개의 작은 페이지 정보들은 페이지 승격과 동시에 작은 페이지 TLB에서 제거 할 수 있기 때문에 작은 페이지 TLB내에 네 개의 엔트리를 다시 사용할 수 있을 뿐만 아니라 큰 페이지 TLB 엔트리 하나로 작은 페이지 TLB 네 개의 엔트리 정보를 가질 수 있기 때문에 시간적 지역성을 보다 효율적으로 이용할 수 있는 장점을 가지고 있다. 예로 가상 주소VA: 00000000, 00001000, 그리고 00002000이 banked-TLB에 저장되어 있고 생성된 가상 주소가VA:00003000이면 banked-TLB와 promotion-TLB를 동시에 참조하게 된다. 만약 두개의 TLB에서 접근 실패가 발생할 경우 MMU가 접근 실패를 처리하는 동안 16KB 경계 (boundary)에 속하는 나머지 세 개의 VPN를 3-cycle에 걸쳐 검색하게 된다. 즉 예처럼 VPN 00000, 00001, 00002이 검색되어지면 promotion-TLB의 하나의 엔트리에 승격이 일어나게 되는데 VPN:00000 에 해당하는 PPN의 정보는 PPN(00)의 위치에 저장되고 VPN:00001에 해당하는 PPN의 정보는 PPN(01), VPN:00002에 해당하는 PPN는 PPN(10), 그리고 접근 실패가 발생된 VPN:00003의 PPN는 PPN(11)의 위치에 각각 저장되어진다. 또한 동시에 banked-TLB에 저장되어 있던 VPN:00000, 000001, 000002의 엔트리는 무효화 시켜 재사용 되도록 한다. 이후 CPU에 의해 가상 주소 VA:00002000 생성되어지면 banked-TLB에서는 접근 실패가 일어나지만 promotion-TLB에서 적중이 일어나게 되고 동시에 네 개의 PPNs중 PPN(10)에 해당하는 PPN만 선택 되어짐에 따라 하나의 PPN만 존재하는 것과 같은 전력 소비를 얻게 된다. 또한 이러한 승격의 동작은 전체 TLB의 접근 실패일 경우에만 일어남으로 일반적으로 접근 실패가 0.1%도 일어나지 않는 TLB의 경우 전력 소비면에서 추가부담 (overhead)으로 거의 적용되지 못한다. 또한 승격을 위한 3-cycle 검색 또한 TLB의 접근 실패 처리 (miss handling) 동안 일어남으로 일반적으로 접근 실패 처리 사이클은 최소한 수-cycle이상임으로 이 또한 추가부담으로 적용되지 못한다.

이러한 구조로 설계될 경우 얻을 수 있는 최대 사상 효과는 작은 페이지 TLB의 엔트리 개수가 m 개, 큰 페이지 TLB의 엔트리 개수가 n 개인 경우 $m+n * (\text{큰 페이지 크기} / \text{작은 페이지 크기})$ 이다. 만약 4KB-16KB 페이지 구성의 경우 $m+4n$ 이며 4KB-32KB 페이지 구성의 경우 $m+8n$ 의 사상 효과를 가질 수 있다. 완전 연관 TLB 구조는 소비 전력인 측면에서 TLB의 엔트리

수가 64개 이상일 경우 기하급수적으로 증가하는 경향을 보이고 있다. 그럼으로 소비 전력을 줄이기 위한 방법중의 하나가 한번에 참조되어지는 TLB의 엔트리 수를 64개 이하로 줄이는 것이 바람직하다. 그러나 높은 성능을 제공하기 위해서는 더 많은 TLB 엔트리를 제공해 주어야 한다. 이러한 상반된 특성을 효과적으로 반영하기 위하여 전체 TLB 공간을 몇 개의 sub-TLBs로 구성하는 बैं크 (bank) 개념을 이용하여 성능 보장과 저전력 효과를 모두 얻고자 하였다. 제안된 방법은 작은 페이지 크기를 지원하는 하나의 완전 연관 TLB를 절반의 엔트리로 구성된 2bank-TLB로 구성하였으며, Bank0-TLB와 Bank1-TLB를 선택적으로 참조함으로써 소비 전력을 절반으로 줄이고자 하였다. 이러한 बैं크의 선택은 생성된 VPN의 최하위 비트를 이용하여 0bit인 경우 Bank0-TLB를 참조하게 되고 1bit인 경우 Bank1-TLB를 참조하게 된다. 또한 bank 구조의 가장 큰 단점인 하나의 बैं크에 편중되는 단점을 보완하면서 성능 향상을 높이기 위하여 promotion-TLB의 동작 메커니즘을 적용하여 고성능과 저전력을 보장할 수 있다. 제안하고자 하는 Banked-promotion TLB는 그림 1와 같다.

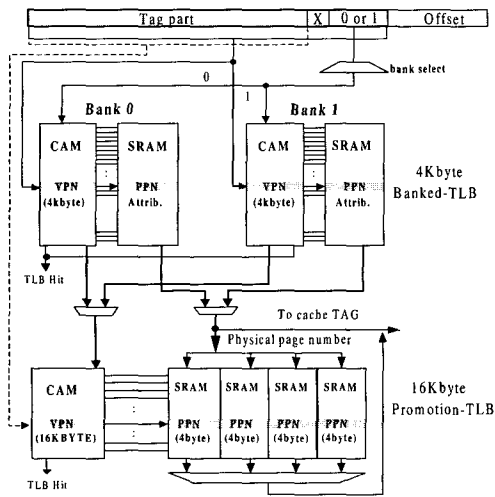


그림 1 Banked-promotion TLB 구조

3.2 BP-TLB 동작 원리

제안된 BP-TLB 동작 원리는 다음과 같다. 먼저 가상 주소가 중앙 처리 장치로부터 발생되면 기본 페이지 크기 (예: 4KB)의 페이지를 찾기 위하여 banked-TLB를 검색하는 동시에 큰 페이지 크기 (예: 16KB)를 검색하기 위하여 promotion-TLB를 동시에 검색한다. 이러

한 검색은 한 사이클에 완료될 수 있다. 이때 세 가지 경우가 가능할 수 있다.

1) bank0-TLB 또는 bank1-TLB에서 접근 성공 (Hit in bank0-TLB or bank1-TLB):

두 개의 बैं크 중 하나의 बैं크를 선택하기 위하여 VPN 최하위 한 비트를 이용하였다. 즉 0인 경우 Bank0-TLB, 1인 경우 Bank1-TLB를 참조하게 된다. 만약 생성된 주소의 적중(hit)이 बैं크에서 일어난다면, 기존의 TLB 적중과 같이 VPN를 PPN으로 변환하여 캐쉬에 보내고 TLB 동작은 끝나게 된다.

2) promotion-TLB에서 접근 성공 (Hit in promotion-TLB):

예로 4KB-16KB 페이지 구성의 경우를 가정하면 16KB VPN의 비트수가 4KB VPN에 비해 2비트 적으므로 이 2비트를 이용하여 4개의 물리적인 페이지 중에 하나만 선별적으로 구동 가능하게 할 수 있다. 4개의 순차적인 작은 가상 페이지가 하나의 큰 페이지 TLB의 엔트리로 승격될 때 PPN(00) 위치에 순차적인 작은 페이지 번호 첫 번째에 해당하는 PPN이 그리고 PPN(11) 위치에 마지막 작은 페이지 번호에 해당하는 PPN이 들어가기 때문에 TLB의 태그 부분인 CAM에서 접근 성공을 결정하는 동시에 하나의 PPN이 결정되어질 수 있다. 이는 전력소모 측면에서도 많은 이득을 볼 수 있다. 만약 banked-TLB에서는 접근 실패지만 promotion-TLB에서 적중이 발생하게 되면 banked-TLB의 적중처럼 변환된 PPN을 캐쉬에 보내고 캐쉬의 태그 비트와 비교 수행을 하게 된다.

3) 두 TLBs에서 접근 실패 (Miss in both places):

- 접근 실패가 발생한 가상 페이지 번호에 해당하는 순차적인 3개의 VPN이 banked-TLB에 존재하지 않는 경우:

먼저 2개의 TLB에서 접근 실패가 발생하면 MMU가 이를 처리하는 동안 접근 실패가 발생한 VPN에 순차적인 3개의 페이지를 검색하기 위하여 접근 실패가 발생한 VPN의 하위 두 비트에 카운터와 모듈러를 이용하여 새로운 VPN를 재구성한 후 작은 페이지 TLB에 대해 재 검색을 수행하여 순차적인 3개의 엔트리를 탐색한 후 만약 3개가 존재하지 않으면 기본 페이지 크기에 해당하는 하나의 PPN이 작은 페이지 TLB내에 새로이 등록이 되며 FIFO 방식으로 채워진다. 이와 같은 가상 페이지 재구성과 재검색은 MMU가 접근 실패를 처리하는 동안 수행되어지기 때문에 부가적인 수행 시간으로 반영되지 않는다. 예로 VA:00000000가 생성되어 지면

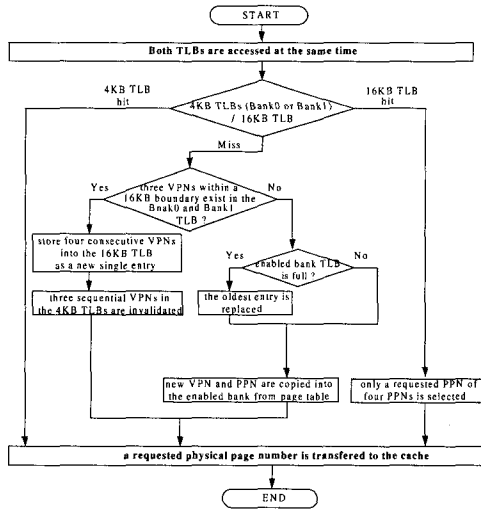


그림 2 Banked-promotion TLB 동작의 흐름도

Bank0-TLB를 검색하기 위하여 VPN: 00000 (20-bits, 16진수), promotion-TLB를 검색하기 위하여 VPN: 00 0000 0000 0000 0000 (18-bit, 2진수)이 생성되고 각각의 TLB 태그와 비교 수행을 하게 된다. 만약 두 개의 TLB에서 모두 접근 실패가 발생하게 되면 실패가 일어난 주소의 16KB 경계에 속하는 나머지 세 개의 VPN를 검색하기 위하여 카운터와 모듈러 (mod 4)를 이용하여 VPN:00001, 00002, 그리고 00003를 생성하여 bank0-TLB와 bank1-TLB를 각각 검색하게 된다. 세 개의 VPN이 bank-TLB에 아직 존재하지 않을 경우 VA:000000000로 접근되어진 Bank0-TLB 하나의 엔트리에 메모리의 페이지 테이블로부터 검색되어진 새로운 PPN이 FIFO방식으로 저장되어진다.

- 접근 실패가 발생한 가상 페이지 번호에 해당하는 순차적인 3개의 VPN이 bank-TLB내에 존재하는 경우:

만약 bank-TLB에서 순차적인3개의 VPN가 존재하게 되면, MMU가 접근 실패를 처리하는 동안 3개의 순차적인 PPN을 FIFO방식으로 promotion-TLB의 한 엔트리를 할당하여 저장시키고 bank-TLB내에 있는 3개의 엔트리를 무효화 (invalid) 시킨다. 그리고 마지막 새로운 PPN이 들어오면 큰 페이지 TLB의 지정된 위치에 저장시키고 TLB의 운용은 끝나게 된다. 이렇게 하나의 큰 페이지로 승격시킴으로 인해 큰 페이지의 효과를 얻을 수 있고 작은 페이지 TLB 엔트리의 4개를 다시 사용할 수 있으므로 물리적으로 제공되는 엔트리의

수를 더 효과적으로 증대시킬 수 있다. 또한 반드시 순차적인 데이터만 승격 가능하게 함으로 미리 저장 공간을 확보하고 있는 기존의 완전 서브블록 TLB (complete subblock-TLB)와의 차별성을 들 수 있다.

제안된 BP-TLB에 대한 동작 흐름도는 그림 2와 같다.

4. 시뮬레이션과 분석적 모델을 통한 성능 평가

시뮬레이션 환경과 성능 평가 지표, 그리고 소비 전력에 대한 다양한 시뮬레이션의 결과가 이 장에서 소개어진다. 벤치마크 프로그램은 SPEC95을 사용하였으며 트레이스 구동 시뮬레이션을 사용하였다. 성능 평가 지표로는 접근 실패율 (miss ratio), 평균 접근 시간 (average memory access time), 소비 전력 (power consumption), 그리고 에너지*지연시간 곱 (energy*delay product)를 사용하였다. 시뮬레이터는 DineroIV 캐쉬 시뮬레이터 [14] 와 CACTI-II 시뮬레이터 [15]를 수정하였다.

4.1 접근 실패율과 평균 메모리 접근 시간 비교

먼저, 접근 실패율의 관점에서 promotion-TLB에 대한 성능 향상 효과를 설명한다. 최적의 페이지 크기와 엔트리 수를 결정하기 위하여 여러 가지 실험을 수행하였다. 시뮬레이션 결과 작은 페이지 TLB의 엔트리 개수가 작은 경우 (예로 32개 이하)에는 작은 페이지 크기는 4KB, 큰 페이지 크기는 8KB가 최적의 성능을 보임을 알 수 있었다. 그러나 최근 대부분의 아키텍처에서 사용되고 있는 기본적인 TLB엔트리 개수인 64 또는 128인 경우, 즉 대략 256KB 또는512KB 메모리 사상 크기 (memory mapping coverage)의 경우에는 작은 페이지 크기로 4KB와 큰 페이지 크기로 16KB의 조합이 최적의 성능을 보임을 알 수 있었다. 시뮬레이션의 결과를 살펴보면 promotion-TLB의 엔트리 수가 훨씬 적음에도 불구하고 같은 메모리 사상 크기를 가지는 완전 연관 TLB와 거의 유사한 성능을 보임을 알 수 있다. 이는 대부분의 벤치마크 프로그램에서 공간적 지역성과 연속성을 반영하는 큰 페이지 TLB의 적중률이 대단히 높기 때문이다. 벤치마크 프로그램을 분석한 결과 하나의 가상주소가 생성되어지면 연속적인 다음 가상 주소가 생성될 확률이 50%이상인 것으로 판명되었다. 따라서 제안된 promotion-TLB 구조는 적은 엔트리 수를 사용함에도 불구하고 높은 성능 향상 효과를 얻을 수 있다.

제안하는 banked-promotion TLB는 저전력-고성능의 효과를 모두 얻기 위하여 제안된 구조이다. 이러한

효과를 비교/분석하기 위하여 크게 두 가지 부류로 나누어 설명하고자 한다. 즉 전통적인 TLB 구조 (traditional TLB) 와 진보된 TLB (advanced TLB)로써 전통적인 TLB 구조로는 기존의 완전 연관 TLB와 micro-TLB를 예로 들었으며, 진보된 TLB 구조로는 victim-TLB와 기존의 banked-TLB 구조를 선택하였다. Micro-TLB의 작은 TLB 와 victim TLB의 주 TLB (main TLB)는 한 사이클로 가정하였으며, 접근 실패가 발생하면 micro-TLB의 L1 TLB와 victim TLB의 victim buffer는 소비 전력을 줄이기 위해 두 사이클로 가정하였다.

그림 3과 그림 4는 전통적인 TLB 구조인 128 엔트리를 가진 완전 연관 TLB와 작은 TLB의 4개 엔트리와 L1 TLB의 128 엔트리를 가진 micro-TLB와의 접근 실패율과 평균 메모리 접근 시간을 보여 주고 있다. 제안된 BP-TLB의 “32-32-16” 사용은 Bank0와 Bank1의 엔트리 개수가 각각 32이며, promotion-TLB의 엔트리 개수가 16개임을 나타낸다. 그림 3에 따르면 BP-TLB가 다른 구조에 비해 가장 높은 접근 실패율을 보이고 있으나 엔트리 개수가 단지 60%만 사용한 결과

이며, 평균 메모리 접근 시간의 관점에서 보면 FA-TLB와 거의 유사한 성능을 보임을 알 수 있다. 여기서는 보여지지 않았지만 같은 면적 비용을 사용한 결과 128 엔트리를 가진 FA-TLB와 비교했을 경우 접근 실패율이 약 90%정도 감소함을 알 수 있었다 [16]. 그림 5와 그림 6은 진보된 TLB 구조와의 성능 비교에 대한 시뮬레이션 결과이다. 이 결과 또한 BP-TLB의 태그 엔트리 수가 48개 적음에도 불구하고 다른 구조에 비해 거의 유사한 성능을 보임을 알 수 있다. 특히, vortex 벤치마크의 경우 한번 참조가 일어난 가상주소에 대해 64개의 엔트리 경계를 넘어 다시 참조되는 경향이 대단히 높기 때문에 상대적으로 전체 Bank 엔트리 수가 64개인 BP-TLB의 성능이 다소 떨어짐을 알 수 있다. 또한 이 벤치마크는 공간적/연속적인 지역성을 거의 보이지 않음으로 promotion-TLB의 사용이 거의 없는 경향을 보임을 알 수 있었다. 시뮬레이션을 수행하기 위한 구체적인 변수 값들은 표 2로써 정의되어진다. 이러한 변수 값들은 일반적인 32-bit 내장형 프로세서에서 사용되어지는 값들을 사용하였다 (예로 Hitachi SH4 or ARM920T).

표 2 시뮬레이션 변수들

CPU clock	200 MHz
Memory latency	15 cpucycle
Memory bandwidth	1.6 Gbytes / sec
Small TLB hit time of Mico-TLB	1 cpucycle
L1 TLB hit time of Mico-TLB	2 cpucycle
Main TLB hit time of victim-TLB	1 cpucycle
Victim buffer hit time of victim-TLB	2 cpucycle

□ FA-128 ▨ Micro-TLB (4-128) ■ BP-TLB (32-32-16)

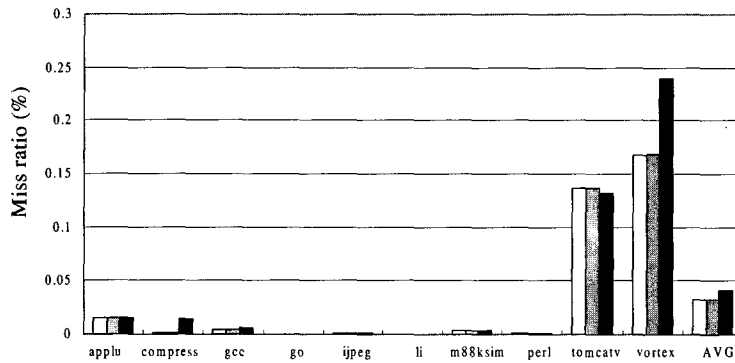


그림 3 BP-TLB 와 전통적인 TLB와의 접근 실패율

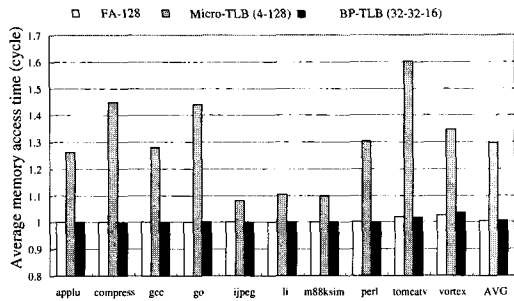


그림 4 BP-TLB 와 전통적인 TLB와의 평균 메모리 접근 시간

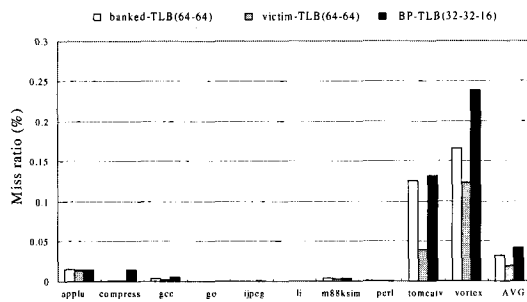


그림 5 BP-TLB 와 진보된 TLB와의 접근 실패율

시뮬레이션 결과에 따르면, micro-TLB와 victim buffer의 2-사이클 오버헤드는 각각 14.2% 그리고 0.08%로 판명되었다. 또한 일반적으로 순수한 banked-TLB의 가장 큰 단점은 생성된 가상주소가 두 개의 bank에 고루 분포되지 못하고 하나의 bank에 편중되어 다른 bank의 활용도가 떨어질 수 있다. 그러나 제안된 BP-TLB는 이러한 구조적 단점들을 큰 페이지를 지원하는 promotion-TLB에 의해 극복할 수 있다.

표 3은 제안된 BP-TLB의 각각의 TLB에서 접근 성공이 일어나는 평균값을 보여준다. 이는 10개의 벤치마크에 대한 평균값이다. 결론적으로, 제안된 BP-TLB는 적은 TLB 엔트리를 사용함에도 불구하고 다른 TLB 구조와 거의 유사한 성능을 보임을 알 수 있다.

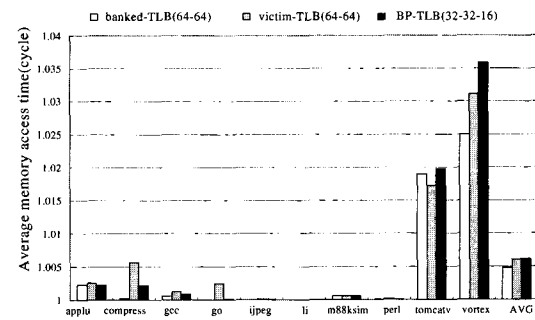


그림 6 BP-TLB 와 진보된 TLB와의 평균 메모리 접근 시간

4.2 BP-TLB 소비전력 비교 (Comparison of BP-TLB power consumption)

완전 연관 TLB는 일반적으로 CAM 셀의 사용으로 모든 엔트리를 동시에 참조하고 비교하게 된다. 그럼으로 소비 전력 측면에서는 이러한 완전 연관 TLB 구조가 가장 적합하지 못한 구조로 알려져 있으나 엔트리가 32 또는 64개 이하인 경우 직접 사상 TLB (direct-mapped TLB) 또는 2-way 집합 연관 TLB (2-way set associative TLB)와 거의 유사한 전력을 소비함을 알 수 있다 [10].

먼저, CACTI-II 시뮬레이터를 [15] 이용하여 다양한 TLB 구조에 대한 시뮬레이션을 수행하였다. CACTI-II 시뮬레이터는 캐쉬 시뮬레이터로 캐쉬를 한번 접근할

표 3 BP-TLB 적중 분포 (Hit proportion)

Benchmark	FA-128	Micro-TLB (4-128)	BP-TLB (32-32-16)
Applu	25.8%	72.2%	2.0%
Compress	49.4%	33.3%	17.4%
Gcc	46.2%	13.3%	40.5%
Go	8.4%	78.6%	12.9%
Ijpeg	7.1%	76.8%	16.1%
Li	14.8%	4.9%	80.3%
M88ksim	6.7%	84.4%	8.9%
Perl	12.5%	25.3%	62.2%
Tomcatv	6.7%	78.1%	15.0%
Vortex	67.1%	25.8%	6.9%
AVG	26.2%	48.6%	25.1%

표 4 다양한 TLB 크기에 대해 한번 참조시 소비되는 소비 전력

# of entries in fully associative TLB	Read / hit (nJ)	Read / miss (nJ)	Write (nJ)
4	2.2630	0.6154	0.2397
16	2.9209	1.1021	0.5326
32	3.7575	1.7511	0.7097
64	5.4200	3.0490	0.9908
128	8.7272	5.6447	1.8551

때 소비되는 소비전력과 접근 시간을 계산할 수 있다. 이 시뮬레이터를 TLB 구조에 맞게 재수정 하였으며, 0.8 mm 공정 기술과 4.5 V사용하였다. 완전 연관 TLB 구조는 대부분의 전력 소비가 각각의 태그 부분을 비교하기 위한 CAM 부분에서 소비되어진다. 표 4는 다양한 크기의 TLB를 한번 참조할 때 적중 / 실패에 따라 소비되는 전력 소모를 보여 주고 있다. 특히 128 엔트리와 64 엔트리의 소비 전력의 큰 차이는 CAM 내부의 매치라인(match line) 과 비트라인 (bit line)에 의해 소비되는 파워가 급격히 증가하기 때문이다. 표 4의 각각의 부분은 TLB 읽기 적중 (read hit), TLB 읽기 실패 (read miss), 그리고 TLB 쓰기 (write) 동작에 의해 소비되는 소비 전력을 보여주고 있다. 일반적으로 TLB는 엔트리의 수에 선형적으로 증가함을 알 수 있다.

표 4을 이용하여 완전 연관 TLB의 평균 소비 전력은 수식 (1) 처럼 구할 수 있다.

$$Avg.power = N_{hit} * P_{hit} + N_{miss} * P_{miss}, \quad (1)$$

여기서 N_{hit} 와 N_{miss} 는 각각 TLB 적중률과 접근 실패율을 나타낸다. 또한 P_{hit} 는 TLB 적중 시 소비되는 소비 전력을 나타내며, P_{miss} 는 TLB 접근 실패 시 소비되는 소비전력을 보여준다. P_{miss} 는 수식 (2) 처럼 계산되어진다.

$$P_{miss} = P_{CAM} + P_{write} + P_{off}, \quad (2)$$

여기서 P_{CAM} 은 TLB의 태그 부분이 참조되어질 때 소비되는 전력이며, P_{write} 는 접근 실패 시 태그 메모리와 데이터 메모리를 업데이트할 때 소비되는 전력이다. P_{off} 는 TLB에서 접근 실패가 발생할 경우 캐쉬와 패드 부분에 의해 소비되는 전력 소비이다. P_{off} 는 수식 (3)처럼 계산되어진다.

$$P_{off} = P_{cache_acc} + M_{cache_miss} * (P_{cache_write} + P_{pad}), \quad (3)$$

여기서 P_{cache_acc} 은 캐쉬 블록을 접근할 때 소비되는 전력이며, M_{cache_miss} 은 캐쉬 접근 실패율이

다. 또한 P_{cache_write} 은 캐쉬 접근 실패 시 캐쉬를 업데이트함으로써 소비되는 전력 소비이며, P_{pad} 은 온-칩 패드 슬롯에서 소비되는 소비전력을 나타낸다. 마지막으로 P_{pad} 은 수식 (4)로 계산할 수 있다.

$$P_{pad} = 0.5 * V_{dd}^2 * (0.5 * (W_{data} + W_{addr})) * 20pF, \quad (4)$$

여기서 W_{data} 와 W_{addr} 는 TLB 접근 실패 시 하위 메모리로 어드레스와 데이터를 보낼 때 이용되는 비트 수로 각각 32-bit를 가정하였다. 오프-칩의 용량성 부하 (capacitive load)는 20pF로 가정하였으며 [17], 또한 데이터 캐쉬 메모리로 32-byte 블록을 지원하는 2-way 집합 연관 캐쉬를 가정하였다. 이러한 다양한 기본적인 변수 값들은 표 5처럼 요약되어진다.

표 5 시뮬레이션 변수 값

M_{cache_miss}	0.05
P_{cache_acc}	21.291 nJ
P_{cache_write}	10.145 nJ
P_{pad}	6.48 nJ
P_{off}	22.122 nJ

그림 7은 기존의 TLB 구조와 BP-TLB의 소비 전력 차를 보여주고 있으며, 그림 8은 전력과 성능을 함께 표현할 수 있는 성능 지표인 에너지*지연시간 곱 (energy*delay product)를 보여주고 있다. BP-TLB의 소비 전력에 대한 결과는 가능한 모든 경우, 즉 두개의 TLB 병렬 참조에 의해 소비되는 전력과 3개의 연속적인 VPNs의 존재 유무와 3개의 PPNs을 promotion-TLB로 승격시킬 때 소비되는 전력을 모두 고려하였다. 그림에서 알 수 있듯이 micro-TLB가 가장 낮은 소비 전력을 보이고 있으나 이는 유사한 성

능을 보이는 완전 연관 TLB의 경우 8개의 엔트리만 사용하고도 micro-TLB의 평균 메모리 접근 시간보다 더 높은 성능을 보이지만 여기서는 128 엔트리를 가진 완전 연관 TLB와 80개 엔트리의 BP-TLB와 비교를 수행하였으므로 이와 같은 결과를 얻을 수 있었다. 그럼으로 낮은 접근 실패율을 보이는 명령어 TLB의 경우에는 micro-TLB 사용에 의미를 가질 수 있으나 데이터 TLB의 경우 높은 성능 저하로 실제 사용에 제약을 받는 구조라 할 수 있다. 그림 8의 결과는 성능과 전력을 모두 고려한 결과로써 제안된 BP-TLB는 완전 연관 TLB에 비해 약 50%이상 성능 향상 효과를 보임을 알 수 있다.

그림 9와 그림 10은 진보된 TLB와 제안된 BP-TLB와의 소비전력과 에너지*지연 시간 곱에 대한 시뮬레이션 결과를 보여주고 있다. 이 결과에서도 제안된 BP-TLB의 소비 전력은 대부분의 벤치마크에서 가장 낮으며 에너지*지연 시간 곱 또한 가장 좋은 성능을 보임을 알 수 있다. 마지막으로 제안된 BP-TLB의 성능 향상 효과를 보기 위하여 성능 향상율(improvement ratio : q)를 사용하였다. 성능 향상율에 대한 식은 다음과 같다.

$$q = ((\alpha - \beta) / \beta * 100\%), \quad (5)$$

여기서 β 는 BP-TLB의 에너지*지연시간 곱의 값을 나타내며, α 는 비교할 다양한 TLB 구조의 에너지*지연시간 곱의 값을 나타낸다.

그림11은 이러한 성능 향상율의 그래프이며 완전 연관 TLB에 비해 약 99.8%, micro-TLB에 비해 약 19.2%, banked-TLB에 비해 약 24.2%, 그리고 victim-TLB에 비해 약 24.4%의 성능 향상을 얻을 수 있었다. 결론적으로 제안된 BP-TLB는 성능적인 면과 소비 전력적인 면을 모두 고려한 새로운 TLB 구조라 할 수 있다.

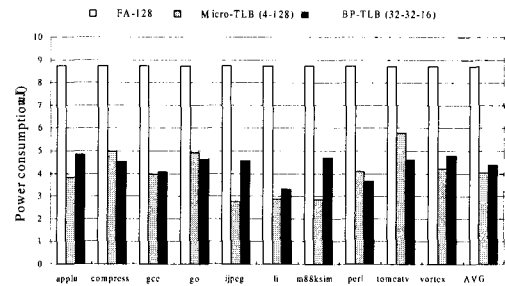


그림 7 BP-TLB와 전통적인 TLB와의 소비전력

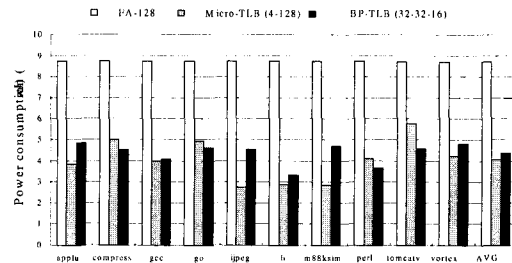


그림 8 BP-TLB 와 전통적인 TLB와의 에너지*지연 시간 곱

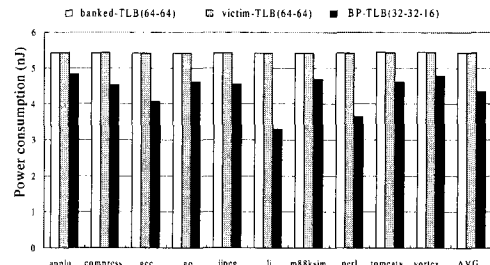


그림 9 BP-TLB 와 진보된 TLB와의 소비전력.

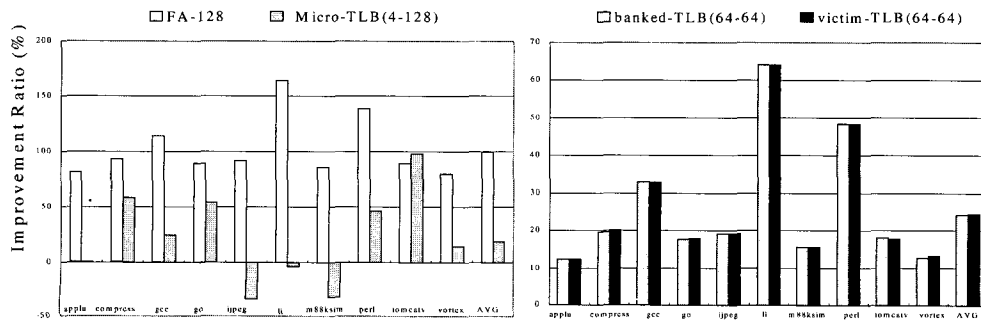


그림 11 다양한 TLBs에 대한BP-TLB의 성능 향상율

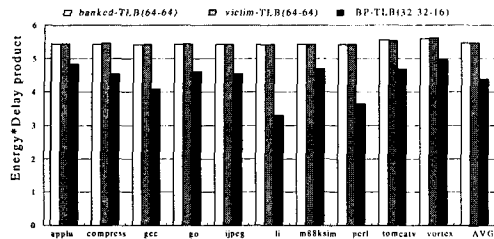


그림 10 BP-TLB 와 진보된 TLB와의 에너지*지연 시간 곱

5. 결론

이 논문의 주목적은 내장형 프로세서의 성능 향상을 위한 새로운 TLB 시스템을 설계하는 것이며 또한 구현이 간단하고 저 전력과 저 비용에 중점을 두었다. 이러한 연구 목적을 달성하기 위하여 뱅크 구조와 이중 페이지 지원을 운영체제의 지원 없이 하드웨어적으로 구현할 수 있는 방법을 제시하였다. 최근의 TLB 연구는 다중 페이지를 지원하는 추세이다. 운영체제의 지원을 받는다면 다중 페이지를 지원하는 것이 TLB 성능을 높이는 데 가장 좋은 방법이지만 아직까지 어떠한 운영체제도 사용자 영역까지 지원하고 있지는 않다. 여러 가지 이유가 존재하지만 특히 외부 단편화 (external fragmentation)가 발생하고 또한 다양한 사용자 프로그램에 대해 운영체제가 페이지 참조 패턴의 정확한 예측이 불가능하기 때문에 동적으로 다중 페이지를 할당할 수 없기 때문이다. 하지만 운영체제의 지원 없이 단지 하드웨어적인 방법으로 이중 페이지를 지원할 경우에도 낮은 가격에 높은 성능을 보일 수 있으며, 또한 TLB내 엔트리 수의 감소와 뱅크의 사용으로 매번 TLB를 접근할 때마다 많은 엔트리를 접근해야 하는 기존의 TLB 구조보다 소모 전력 면에서도 상당히 유리하다. 결론적으로 제안하고자 하는 TLB는 저전력-고성능의 효과를 극대화 시키기 위한 방법으로 기존의 내장형 프로세서에서 널리 사용되고 있는 완전 연관 TLB에 비해서는 성능과 소비전력을 약 100%정도 향상시키는 결과를 얻을 수 있었다. 또한 이 기술은 구현이 간단하다는 장점을 가지며 운영체제의 수정 없이 다양한 TLB 구조에 접목이 가능하다는 장점을 가진다.

참고 문헌

[1] Todd M. Austin and Gurindar S. Sohi, "High-bandwidth address translation for multiple-issue

processors," In *proceedings of the 32rd ACM Intl Symp. on Computer Architecture*, pp. 158-167, May 1996.

- [2] M. Talluri, Shing Kong, Mark D. Hill, and David A. Patterson, "Tradeoffs in Supporting Two Page Sizes," In *Proc. the 19th Annual Intl Symp. on Computer Architecture*, pp. 415-424, May 1992.
- [3] Y. A. Khalidi, "Virtual memory support for multiple page sizes," In *Proc. of the fourth workshop on workstation operating systems*, Oct. 1993.
- [4] Sribalan Santhanm, "StrongARM SA110, a 160mhz 32b 0.5w CMOS ARM processor," In *Hot Chips 8*, Aug. 1996.
- [5] M. B. Kamble and K. Ghose, "Energy-Efficiency of VLSI Cache: A Comparative Study," in *Proc. IEEE 10-th Intl. Conf. On VLSI Design*, pp. 261-267, Jan. 1997.
- [6] M. B. Kamble and K. Ghose, "Analytical Energy Dissipation Models for Low Power Caches," *ACM/IEEE Intl Symp. on Low-Power Electronics and Design*, Aug. 1997.
- [7] Ghose, K. and Kamble, M.B., "Reducing power in superscalar processor caches using subbanking, multiple line buffers and bit-line segmentation," *ACM/IEEE Intl Symp. on Low-Power Electronics and Design*, pp. 70-75, Aug. 1999.
- [8] Kin, et. al., "Filtering memory references to increase energy efficiency," *IEEE Transactions on Computers*, Vol. 49, No. 1, January 2000.
- [9] D. Liu, and C. Svensson, "Trading Speed for Low Power by Choice of Supply and Threshold Voltages," *IEEE journal of solid state Circuits*, Vol. 28, No. 1, 1993.
- [10] T. Juan, T. Lang, J. Navarro, "Reducing TLB Power Requirements," *Intl Symp. on Low Power Electronics and design*, 1997.
- [11] Mark D Hill and M. Talluri. "Suppassing the TLB Performance of Superpages with Less Operating System Support," *ASPLOS VI-, San Jose, California USA*, pp. 171-182, Oct. 1994.
- [12] J. Kin, M. Gupta, and W. H. Mangione-Smith, "The Filter Cache: An Energy Efficient Memory Structure," *MICRO-97: ACM/IEEE International Symposium on Microarchitecture*, Research Triangle Park, NC, pp. 184-193, Dec. 1997.
- [13] S. Manne, A. Klausner, D. Grunwald, F. Somenzi, "Low power TLB Design for High Performance Microprocessors," *Univ. of Colorado Technical Report*, 1997.
- [14] Jan Edler and Mark D. Hill, "Dinero IV

- Trace-Driven Uniprocessor Cache Simulator," available from Univ. Wis., CS ftp site 1997.
- [15] Glenn Reinman and Norm Jouppi, "An Integrated Cache Timing and Power Model," *Compaq WRL Report*, 1999.
- [16] J. H. Lee, J. S. Lee, G. H. Park, G. W. Lee, and S. D. Kim, Dual TLB structure to support two page sizes, *Electronics Letters*, Vol. 36, No. 8, pp. 705-706, Apr. 2000.
- [17] S. J. E. Wilton, and N. Jouppi, An Enhanced Access and Cycle Time Model for On-Chip Caches, *Digital WRL Research Report 93/5*, July 1994.



이 정 훈

1999년 성균관대학교 제어계측공학과(학사). 2001년 연세대학교 컴퓨터과학과(석사). 2001년 ~ 현재 연세대학교 컴퓨터과학과(박사과정). 관심분야는 지능형 메모리 시스템, 고성능 컴퓨터구조, 프로세서-메모리 집적구조, 내장형 프로세서

설계임



김 신 탁

1982년 연세대학교 공과대학 전자공학과(학사). 1987년 University of Oklahoma 전기공학(석사). 1991년 Purdue University 전기공학(박사). 1993년 2월 ~ 1995년 2월 광운대학교 컴퓨터공학과 조교수. 1995년 3월 ~ 현재 연세대학교 공과대학 컴퓨터과학과 부교수. 관심분야는 병렬처리 시스템, 컴퓨터구조, Heterogeneous computing임.