연속적 접근 판별 알고리즘을 이용한 저전력 TLB 구조

이 정 훈†

요 약

본 논문은 내장형 프로세서의 소비 전력을 줄이기 위한 저전력 TLB 구조를 제안하고자 한다. 제안된 TLB는 다수의 뱅크로 구성되어지며, 각각의 뱅크들은 하나의 블록 버퍼와 하나의 비교기를 포함한다. 블록 버퍼와 메인 뱅크는 특정 비트를 이용하여 선택적으로 접근이 가능하다. 그러므로 필터링 구조처럼 블록 버퍼에서 적중이 발생하면 메인 TLB 뱅크의 구동 소비 전력이 없고 단지 하나의 엔트리로 구성된 블록 버퍼에 의한 소비 전력만 발생함으로써 소비 전력을 효과적으로 줄일 수 있다. 또한 다른 계층적 구조와는 달리 이중 사이클에 대한 오버헤드가 1%로써 거의 무시 가능하다. 이에 반해 대표적인 계층 구조인 필터 구조의 경우 대략 5%이상 발생하게 되며, 제안된 구조와 동일한 구조를 가지지만 연속적 접근 판별 알고리즘을 사용하지 않은 동일한 구조의 블록 버퍼-뱅크 구조의 경우 15% 이상의 이중 사이클 오버헤드가 발생하게 된다. 이러한 이중 사이클은 프로세서의 성능 저하를 초래함으로써 데이터의 경우 특히 적용이 어려운 단점으로 지적되었다. 소비 전력의 감소 효과는 기존 완전 연관 구조에 비해 95%, 필터 구조에 비해 90%, 연속적 접근 판별 알고리즘 사용하지 않은 동일 구조에 비해 40%의 소비 전력 감소 효과를 얻을 수 있다.

키워드 : 메모리 계층구조, 메모리 관리, 주소변환버퍼(TLB), 저전력, 성능 평가시뮬레이션

Low Power TLB System by Using Continuous Accessing Distinction Algorithm

Jung-Hoon Lee†

ABSTRACT

In this paper we present a translation lookaside buffer (TLB) system with low power consumption for embedded processors. The proposed TLB is constructed as multiple banks, each with an associated block buffer and a corresponding comparator. Either the block buffer or the main bank is selectively accessed on the basis of two bits in the block buffer (tag buffer). Dynamic power savings are achieved by reducing the number of entries accessed in parallel, as a result of using the tag buffer as a filtering mechanism. The performance overhead of the proposed TLB is negligible compared with other hierarchical TLB structures. For example, the two-cycle overhead of the proposed TLB is only about 1%, as compared with 5% overhead for a filter (micro)-TLB and 14% overhead for a same structure without continuos accessing distinction algorithm. We show that the average hit ratios of the block buffers and the main banks of the proposed TLB are 95% and 5% respectively. Dynamic power is reduced by about 95% with respect to with a fully associative TLB, 90% with respect to a filter-TLB, and 40% relative to a same structure without continuos accessing distinction algorithm.

Key Words : Memory Hierarchy, Memory Management, TIB, Low Power, Performance Evaluation

1. 서 론

현재는 다양한 모바일 단말기를 중심으로 멀티미디어 및 통신 응용분야 등 다양한 형태의 내장형 시스템이 보급되고 있다. 특히 와이브로 단말기, HSDPA(High Speed Downlink Packet Access) 폰, 네비게이션, PDA(personal digital assistance)와 같은 고성능 포터블 시스템의 보급은 범용의 마이크로프로세서 대신 고성능/저전력 내장형 프로세서에 대한 요구를 증대시키고 있다. 오늘날 컴퓨터 시스템내의 다양한 분야 중에서 TLB(translation look-aside buffer)는 캐쉬메모리와 함께 가상 메모리 지원 및 메모리 접근 지연시간(memory access latency)과 소비 전력을 줄이고 전체 시스템의 성능 향상을 높이기 위한 필수적인 하드웨어로써 구현되고 있다. TLB는 가상 주소(virtual addresses)를 물리 주소(physical addresses)로 변환하기 위한 페이지 테이블(page table)을 구성하는 캐쉬 메모리이다[1]. 만약 필요로 하는 변환 정보가 TLB내에 존재 한다면, 시스템은 주어진 가상 주소를 페이지 테이블에 접근 하지 않고 적절한 물리 주소로 변환할 수 있다. 하지만 변환 정보가 TLB내에 존재

† 정 회 원 : 경상대학교 전기전자공학부 공학연구원 조교수
논문접수 : 2005년 10월 31일, 심사완료 : 2006년 7월 11일

하지 않는다면, 페이지 테이블을 다시 참조하고 필요로 하는 정보를 TLB로 업데이트 시켜줘야 한다. TLB는 전형적으로 모든 명령어와 데이터 인출 시에 접근해야 하는 메모리 구조로써 접근실패율이 가장 낮은 완전 연관 (fully-associative) 구조로 구성되어 있다. TLB의 CAM(Content Addressable Memory) 과 SRAM의 데이터 부분들이 동적(dynamic) 회로로 구성되어 있기 때문에, 그들은 많은 양의 파워를 소비한다. 뿐만 아니라 TLB 회로들은 임계 시간 경로들로 이루어져 있기 때문에, 회로 지연을 통하여 소비 전력을 절약하는 이점을 얻을 수 없다.

일반적으로 TLB는 매우 작은 사이즈임에도 불구하고 전체 칩에서 소모되는 전력의 상당부분을 차지한다. 예를 들어, 대표적인 RISC 내장형 프로세서인 StrongARM-110[2]과 Hitachi's SH-3[3]의 전력 소모를 살펴보면, 전체 칩의 소비 전력중 TLB가 차지하는 부분이 대략 17%인 것으로 나타나 있다. ARM-920T[4]의 경우 전체 칩의 약 10%을 차지한다. 작은 용량을 가진 메모리임에도 불구하고 TLB가 많은 전력 소모를 보이는 이유는 다음과 같다. 첫째, 온 칩 메모리 시스템을 구성하는 태그와 데이터 배열들은 프로세서의 빠른 클락 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 클락 사이클마다 접근되어야 한다. 셋째, 이러한 메모리 시스템 접근 시에 발생 할 수 있는 접근 실패(miss)는 또 다른 대 용량의 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 가장 기본적인 접근 방법이 된다. 이처럼 TLB가 캐쉬와 더불어 전력 소모가 많은 IP 블록임에도 불구하고 지금까지 저전력 메모리 시스템에 대한 연구는 주로 캐쉬 메모리에 초점을 맞추고 진행되어 왔다[5-8].

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[5-8] 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 설계비용으로 큰 효과를 도출시킬 수 있으며, 저전력 설계 중 가장 포괄적인 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 소비전력 감소율은 완전 연관 TLB에 비해 약 95%, 필터 TLB에 비해 약 90%, 제안된 알고리즘을 사용하지 않은 동일 구조 TLB에 비해 약 40%의 성능 향상을 얻을 수 있었다.

이 논문의 나머지 부분은 다음과 같다. 관련 연구는 제 2장에서 소개되어지며, 제 3장은 제안된 TLB의 구조와 동작 원리에 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 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 실패시 처리해야 하는 지연시간은 대단히 높으므로 고성능을 보장하는 완전 연관 구조가 적합하다고 할 수 있다.

저전력의 효과를 얻기 위한 TLB 구조로는 계층적인 필터 TLB(filter-TLB) 구조[11]가 일반적으로 이용되고 있다. 필터 TLB 구조는 주 TLB 보다 상위 계층에 작은 크기의 TLB를 운영하는 방법으로, 소비 전력의 측면에서는 효과적인 구조이지만 데이터 TLB의 경우 낮은 성능으로 일반적으로 명령어 TLB에 국한되어 사용되고 있다. 예로 SH4 와 PA-RISC2.0는 4개 엔트리로 구성된 필터 명령어 TLB를 단일 TLB(Unified-TLB) 상위 계층에 위치시켜 명령어에 대해서 한 사이클 내에 참조가 일어나고, 참조 실패인 경우 다음 사이클 동안 단일 TLB를 참조하는 메커니즘을 사용하고 있다. 또한 최근의 ARM11[12]에서도 이러한 필터 TLB 구조를 명령어 TLB에 사용하여 소비 전력을 줄이고자 하였다. Ghose[13]는 기존의 필터링 구조의 단점을 극복하기 위하여 라인 버퍼링(line-buffering) 구조를 제안하였다. 즉 필터 버퍼 구조에서 먼저 접근하는 필터링 메커니즘과는 달리 필터 버퍼와 주 메모리 구조를 동시에 접근하는 방법을 사용하여 이중 접근 사이클을 줄이고자 하였으나 소비 전력 면에서는 필터 구조가 더 효과적인 메커니즘이라 할 수 있다. 제안하는 TLB 구조는 이러한 필터 및 라인 버퍼링의 단점을 극복하면서도 성능 및 소비 전력을 모두 줄일 수 있는 새로운 방법을 제시하고자 한다.

이외에도 저전력 TLB를 위하여 CAM 메모리 셀 자체를 변형시키는 방법[10], 낮은 공급 전압을 이용하는 방법[9], 그리고 뱅크 구조를 이용한 방법들이 있다[14]. 메모리 셀 자체를 변화시키는 것은 하드웨어 비용이 증가하는 단점을 가지게 되고, 낮은 공급 전압을 제공하는 방법은 다른 기술

적인 문제를 해결해야하는 어려운 작업이라 할 수 있다. 또한 순수한 뱅크 구조는 저전력 TLB에는 효과적이지만 하나의 뱅크에 편중될 확률이 높으므로 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.

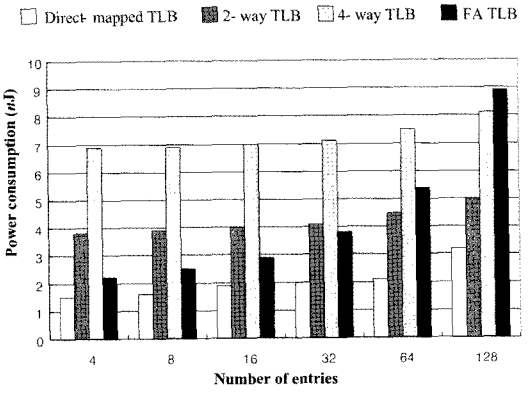
3. 제안된 TLB의 구조적 특징과 동작 원리

이 장에서는 제안된 TLB의 동작 모델과 구조적 특징을 설명하고 새로운 TLB 시스템을 제안하게 된 배경과 동기에 대해서 상세히 살펴 볼 것이다.

3.1 제안된 TLB 시스템 구조

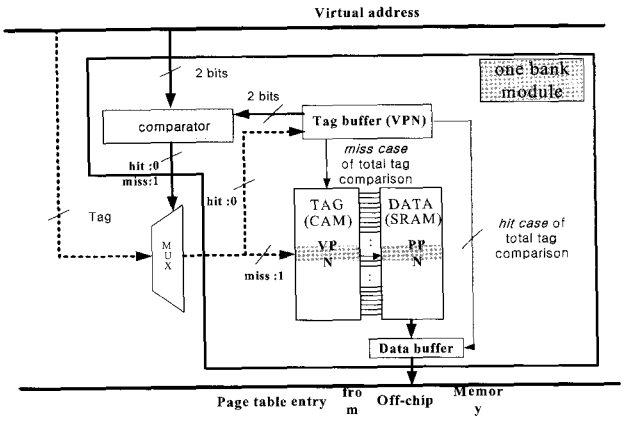
이 연구의 주목적은 실제 구현이 간단하면서 저전력과 높은 성능 향상 효과를 얻을 수 있는 TLB 시스템을 설계하는 것이다. 완전 연관 TLB의 태그 부분은 메모리 참조로부터 얻은 태그와 모든 태그 엔트리들의 병렬적인 수행을 비교하는 추가적인 트랜지스터로 이루어진 CAM(content addressable memory)을 사용하여 구현된다. 만약 임의의 엔트리에 포함된 태그가 비트 라인을 통하여 입력된 태그와 일치한다면 그것은 응답 라인이기 때문에 전압을 인가해준다. 그리고 모든 다른 라인들은 전압을 인가해주지 않는다. 이때 선택된 매치 라인만이 SRAM의 연관된 워드라인을 활성화 시킨다. 완전 연관 TLB의 구조는 추가적인 비교 로직이나 멀티플렉서가 필요하지 않다. 그러나 그것의 접근 시간은 태그 비교가 SRAM으로부터 데이터를 읽어 들임과 동시에 수행되지 않기 때문에 다른 구조의 TLB보다 접근 시간이 오래 걸린다. 게다가 CAM에 대한 각각의 접근을 위하여 모든 라인이 미리 전압 충전(precharge)되어 있어야 하며 매치 신호를 생산하지 못하는 모든 매치 라인은 그때 방전(discharge)되어야 한다. 이러한 전압 충전과 방전은 TLB의 에너지 분배에 중요한 부분으로 작용한다. 특히 완전 연관 TLB 구조는 엔트리 개수가 64개를 넘을 경우 (그림 1)처럼 갑자기 파워 소비가 증가하는 경향을 보인다. 그러므로 소비 전력을 줄이기 위하여 TLB 엔트리 수는 64개 또는 32보다 적어야 한다. 이에 대한 0.13*um* 공정 기술에 대한 다양한 TLB 구조의 소비 전력은 (그림 1)과 같다. 그러나 고성능을 보장하기 위하여 더욱 많은 엔트리를 공급하는 것이 이상적이다. 우리는 한 번에 접근되어지는 TLB 엔트리 수를 낮추고 성능 저하를 줄이기 위하여 뱅크 구조를 기본 구조로 채택하였다. 이는 많은 시뮬레이션 수행을 통하여 가장 효율적인 뱅크의 수가 4개라는 사실을 확인할 수 있었다.

(그림 2)는 제안하는 연속적 접근 판별 알고리즘을 이용한 TLB 구조를 나타내고 있다. 그림을 단순화시키기 위하여 하나의 뱅크로 구성된 형태를 보이지만 제안된 연속적 접근 판별 알고리즘을 이용한 TLB는 4개의 뱅크 모듈로 구성되어 있고, 하나의 뱅크 모듈마다 두 비트 비교기, 하나의

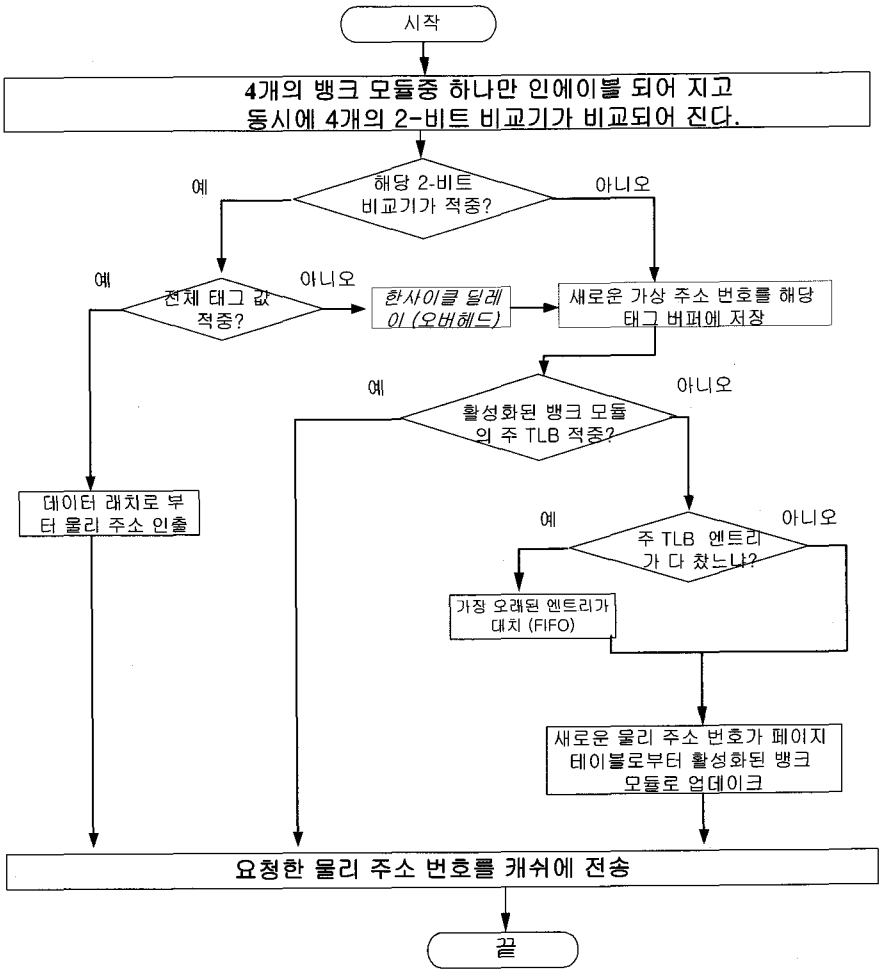


(그림 1) 다양한 TLB 구조에 대한 소비 전력

태그 버퍼, 16-엔트리로 구성된 뱅크 TLB, 그리고 출력 버퍼로 구성된다. 그리고 전체 TLB 구조에서 전체 태그값를 비교하기위한 20-비트 비교기가 하나 추가되어진다. 하나의 뱅크 모듈에 대한 선택은 가상 페이지 번호(virtual page number)의 최하위 두 비트를 이용하여 선택하였다. 즉, 두 비트가 00비트이면 첫 번째 뱅크가 접근되어지며, 11비트이면 네 번째 뱅크가 접근되어진다. 각각의 뱅크와 연관된 태그 버퍼는 부합된 뱅크 TLB 모듈내에 속해있는 가장 최근에 접근된 가상 페이지 번호를 위한 태그 값을 저장한다. 그러므로 4개의 태그 버퍼는 가장 최근에 사용되어진 4개의 가상 페이지 번호가 저장되어진 형태를 취하고 있다. 2-비트 비교기는 CPU로부터 새롭게 생성된 가상 주소 페이지 번호의 두 비트와 태그 버퍼 내에 존재하는 가상 페이지 번호의 두 비트를 비교한다. 이러한 두 비트 비교기 동작 시간은 4개의 뱅크 모듈중 하나의 뱅크 모듈를 선택하기 위한 멀티플렉스(MUX) 시간동안 수행되기 때문에 추가적인 비교 로직에 따른 지연시간은 발생하지 않는다. 시뮬레이션에서 비교를 위해 사용되는 특별한 두 비트는 가상 페이지 번호의 하위 4, 5번째의 두 비트이다. 이는 대부분의 프로그램 수행 집합(working set)이 32KB 또는 64KB 범위 내에서 동작하므로 이러한 두 비트(4, 5번째 비트) 비교가 가장 높은 정확도를 보인다. 만약 더욱 많은 비트가 비교를 위해



(그림 2) 최근 접근된 데이터를 효과적으로 이용 가능한 TLB 구조



(그림 3) 연속적 접근 판별 알고리즘을 가진 TLB 동작의 흐름도

사용된다면 더 높은 정확성을 얻을 수 있으나 비교 시간과 하드웨어 비용에 따른 오버헤드는 증가할 것이다. 구체적인 동작 원리는 다음 절에서 자세히 설명한다.

3.2 제안된 TLB 구조의 동작 원리

제안된 TLB 동작 원리는 다음과 같다. 먼저 가상 주소가 중앙 처리 장치로부터 발생되면 4개의 뱅크 모듈중 하나의 뱅크 모듈을 선택하기 위하여 MUX 동작이 수행된다. 이와 동시에 4개의 태그 버퍼에 저장되어 있는 태그값들의 4, 5번째 비트와 새로 생성된 가상 페이지 번호의 4, 5번째 비트가 두-비트 비교기를 통하여 동시에 병렬적으로 비교되어 진다. 이때 활성화된 뱅크 모듈에 해당하는 두-비트 비교기의 결과에 따라 다음과 같은 동작이 수행되어진다. 참고로 뱅크 모듈 활성화후 두-비트 비교도 가능하나 접근 시간을 줄이기 위하여 뱅크 모듈 선택과 두-비트 비교를 동시에 수행하였다.

3.2.1 선택되어진 뱅크 모듈의 두-비트 비교기가 미스 (miss)인 경우

두-비트 비교기에서 미스가 발생하면, 그것은 새로 생성된 가상 페이지 번호가 명확히 태그 버퍼 안에 있지 않다는 것을 의미한다. 그러므로 버퍼 내에 존재하는 전체 태그 값을 비교할 필요도 없이 바로 한 사이클 동안 활성화된 TLB 뱅크를 탐색하게 된다. 이와 동시에 태그 버퍼는 연속적으로 동일한 가상 주소 생성에 반응하기 위하여 생성된 가상 페이지 번호를 태그 버퍼에 업데이트 하게 된다. 이는 다음 생성 주소에 동일한 가상 주소가 바로 생성되어지면 주 TLB 뱅크 탐색 없이 낮은 소비 전력을 가지는 하나의 버퍼만을 접근함으로써 소비 전력을 줄일 수 있다.

주 TLB 뱅크 탐색 후 적중 (hit)하게 되면 기존의 TLB 동작처럼 요청한 물리 페이지 주소 (physical page number)를 캐쉬에 보내주고 동작을 마치게 되며, 만약 주 TLB에서도 미스가 발생하게 되면 MMU (memory management unit)에 의해 메모리 내에 존재하는 페이지 테이블을 검색하여 캐쉬에 물리 페이지 번호를 전송함과 동시에 TLB에 새로 저장하게 된다.

3.2.2 선택되어진 뱅크 모듈의 두-비트 비교기가 적중 (hit)인 경우

두-비트 비교기에서 적중이 발생하면 이는 새로 생성된

가상 페이지 번호와 태그 버퍼에 존재하는 가상 페이지 번호가 일치할 확률이 대단히 높다. 그러므로 한 사이클 동안 단지 태그 버퍼 내에 존재하는 가상 페이지 번호와 새로 생성된 가상 페이지 번호를 비교하게 된다. 만약 태그 버퍼내의 가상 페이지 번호와 새롭게 생성된 가상 페이지 번호가 일치한다면 같은 뱅크 모듈에 연속적으로 동일한 주소를 접근한 경우임으로 출력 드라이버 (data buffer)에 앞서 접근시 미리 나와 있던 물리 페이지 번호를 주 TLB 탐색 없이 바로 캐쉬로 보내어 줄 수 있다. 이는 동일한 뱅크 모듈에 대하여 연속적으로 동일한 가상 페이지 번호가 생성된 경우로 단지 하나의 버퍼를 구동하는 소비 전력만 사용되어 전체 소비 전력을 효과적으로 줄일 수 있다. 다음 장에서 자세히 설명하겠지만 시뮬레이션 결과에 따르면 버퍼 적중률이 90% 이상으로써, 주 TLB의 소비 전력을 90%이상 감소시키는 효과를 얻을 수 있다.

만약 두-비트 비교기에서 적중이 발생하였으나 전체 태그 비교 시 미스가 발생하게 되면 한 사이클 추가적인 오버헤드가 발생하게 되고, 다음 사이클 동안 주 TLB 뱅크를 탐색하게 된다. 이후의 동작은 기존의 TLB 동작과 동일하다. 그러므로 핵심 사항은 두 비트 비교기에서 적중하였을 경우 전체 태그 값 비교 또한 적중하는 것이 소비 전력 및 성능에 지배적 영향을 미치게 되는 것이다. 이에 가상 페이지 번호 4, 5번째 비교가 가장 좋은 결과를 보임을 많은 시뮬레이션을 통해 알 수 있었다. 또한 다른 필터링 알고리즘처럼 매번 버퍼를 탐색하고 미스 발생시 주 TLB를 탐색하는 오버헤드를 선택적 알고리즘을 이용하여 선택적으로 버퍼 탐색 및 주 TLB 탐색을 한 사이클에 수행함으로써 두 사이클 오버헤드를 획기적으로 줄이는 결과를 얻을 수 있으며, 버퍼의 적중률을 높임으로써 소비 전력을 효과적으로 줄일 수 있는 구조라 할 수 있다. 제안된 TLB에 대한 동작 흐름도는 (그림 3)과 같다.

4. 시뮬레이션과 분석적 모델을 통한 성능 평가

시뮬레이션 환경과 성능 평가 지표, 그리고 소비 전력에 대한 다양한 시뮬레이션의 결과가 이 장에서 소개되어진다. 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)을 사용하였다. 시뮬레이터는 DineroIV 시뮬레이터[15] 와 CACTI-II 시뮬레이터[16]를 수정하였다. <표 1>은 시뮬레이션을 위한 기본 변수 값이다. 이러한 값은 일반적인 32 비트 프로세서의 기본 값이다. (예, ARM 920T)

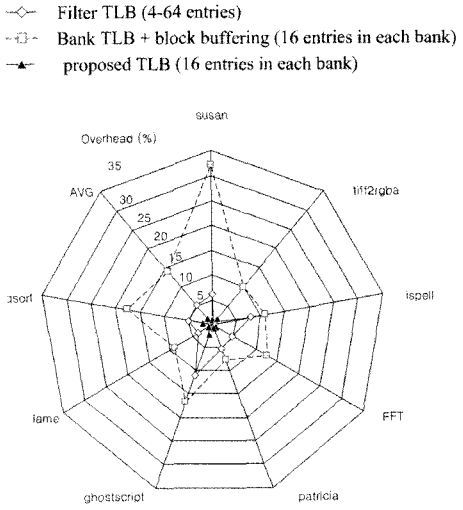
〈표 1〉 시뮬레이션 변수들

변수들	값
CPU clock	400 MHz
Memory clock	266Mhz
Memory latency	35ns

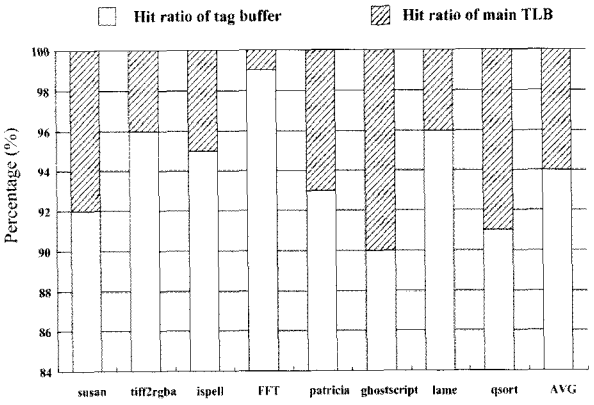
4.1 선택적 알고리즘의 정확도와 오버헤드

디자인 변수와 크기 등을 고려하여 많은 시뮬레이션을 수행하였다. 특히 생성되어진 가상 주소 번호와 태그 버퍼에 있는 가상 주소 번호의 일치성을 높이기 위한 특별한 두 비트를 검출하기 위하여 다양한 실험을 수행하였다. 시뮬레이션 결과에 따르면 가상 주소 번호의 최하위 네 번째 비트와 다섯 번째 비트를 비교하는 것이 정확도를 높이는데 가장 좋은 결과를 보였다. 물론 세 비트 이상 비교시 더욱 높은 정확도를 얻을 수 있었으나 추가적인 오버헤드와 지연시간 등을 고려할 때 두 비트 비교가 가장 효과적인 구현 방법이라 할 수 있다.

제안된 구조의 가장 효과적인 방법은 두 비트 비교 적중시 생성된 가상 주소 번호와 태그 버퍼 내에 존재하는 가상 주소 번호의 일치성을 높이는 것이다. 만약 가상 주소 번호가 서로 일치하지 않으면 추가적인 한 사이클 오버헤드가 발생하게 되고 이것은 성능에 직접적인 영향을 미치게 된다. (그림 4)는 이러한 추가적인 한 사이클 오버헤드에 대한 시뮬레이션 결과이다. 비교를 위하여 동일한 엔트리를 가진 필터 TLB와 선택적 알고리즘을 사용하지 않은 동일한 구조의 뱅크 모듈 구조가 사용되었다. 필터 구조의 경우 추가적



(그림 4) 추가적인 오버헤드



(그림 5) 제안된 구조의 태그 버퍼와 주 TLB의 적중 분포율

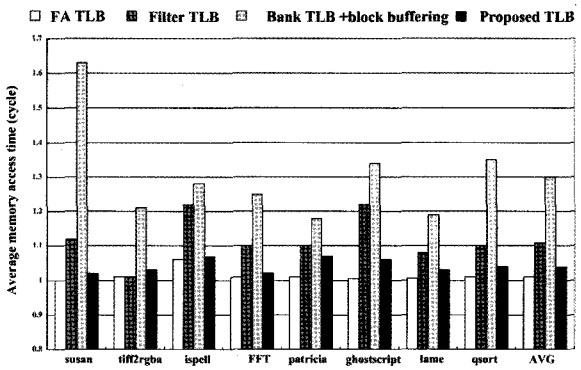
인 사이클 오버헤드는 약 5%를 나타내고 있으며, 연속적 접근 판별을 사용하지 않은 동일한 구조의 뱅크 모듈은 약 15% 이상 발생하였다. 그러나 제안된 구조의 경우 대부분의 벤치마크에 대해서 약 1%로써 성능 저하를 극소화 시킬 수 있다.

(그림 5)는 제안된 구조에서 뱅크 모듈의 태그 버퍼 적중률과 주 뱅크 TLB 적중률을 보여주고 있다. 대부분의 미디어 벤치마크에서 평균 94%의 태그 버퍼에서 적중되고 있음을 알 수 있다. 이는 주 TLB 부분을 약 94% 접근하지 않음으로써 소비 전력의 효과를 극대화할 수 있으며, 비동기 회로로 구성할 경우 더욱 빠른 접근 시간을 제공해 줄 수 있다.

4.2 접근 실패율과 평균 메모리 접근 시간

다양한 TLB 구조와 제안된 TLB 구조의 성능 비교를 위하여 접근 실패율과 평균 메모리 접근 시간에 대하여 설명한다. CACTI-II[16] 시뮬레이션에 따르면 접근되어지는 TLB 엔트리 개수에 따라 접근시간이 많이 달라짐을 알 수 있다. 즉 하나의 엔트리로 구성된 버퍼 엔트리 접근 시간이 4개의 엔트리로 구성된 필터 TLB 접근 시간에 비해 두 배 이상 빠른 접근시간을 보이고 있다. 비동기 회로(asynchronous circuit)로 구성할 경우, 이러한 계층 구조로 구성하였을 때 더욱 효과적인 접근 시간을 얻을 수 있다. 그러나 여기서는 모두 한 사이클로 가정하여 시뮬레이션을 수행하였다.

(그림 6)은 다양한 TLB 구조, 예로 64개의 엔트리로 구성된 완전 연관 TLB (FA TLB), 4개의 필터 엔트리를 가진 필터 TLB (filter TLB), 4개의 뱅크 모듈 위에 하나의 버퍼를 가진 버퍼-뱅크 구조 (bank-TLB + block buffering), 그리고 제안된 연속적 접근 판별 알고리즘을 가진 선택적 블록 버퍼 뱅크 구조 (proposed TLB)에 대한 평균메모리 접근 시간을 보여주고 있다. 완전 연관 TLB를 제외한 나머지는 모두 4개의 엔트리를 추가적으로 가지고 있다. 시뮬레이션 결과에 대한 차이는 한 사이클 추가적인 오버헤드에 대한 결과 때문이다. 동일한 구조로 구성하였지만 연속적 접근 판별 알고리즘을 사용한 구조와 사용하지 않은 구조의 결과가 성능 결과에 매우 영향을 미치고 있음을 알 수 있다. 또한 저전력에 매우 효과적인 구조인 필터 TLB 구조 역시 성능 저하가 매우 큼을 알 수 있다. 그러므로 제안된 연속적 접근 판별 알고리즘을 이용한 TLB 구조는 성능의 저하가 매우 적으면서 소비 전력에도 효과적인 구조라 할 수 있다.



(그림 6) 다양한 TLB 구조에 대한 평균 메모리 접근 시간

4.3 소비 전력 비교

완전 연관 TLB 구조는 대부분의 전력 소비가 각각의 태그 부분을 비교하기 위한 CAM 부분에서 소비되어진다. 일반적으로 TLB는 엔트리의 수에 선형적으로 증가함을 알 수 있다.

완전 연관 TLB의 평균 소비 전력은 수식 (1)처럼 구할 수 있다.

$$Avg.power = Nhit * Phit + Nmiss * Pmiss , \quad (1)$$

여기서 *Nhit* 와 *Nmiss* 는 각각 TLB 적중률과 접근 실패율을 나타낸다. 또한 *Phit*는 TLB 적중 시 소비되는 소비 전력을 나타내며, *Pmiss*는 TLB 접근 실패시 소비되는 소비 전력을 보여준다. *Pmiss*는 수식 (2) 처럼 계산되어진다.

$$Pmiss = PCAM + Pwrite + Poff , \quad (2)$$

여기서 *PCAM* 은 TLB의 태그 부분이 참조되어질 때 소비되는 전력이며, *Pwrite* 는 접근 실패 시 태그 메모리와 데이터 메모리를 업데이트할 때 소비되는 전력이다. *Poff*는 TLB에서 접근 실패가 발생할 경우 캐쉬와 패드 부분에 의해 소비되는 전력 소비이다. *Poff*는 수식 (3)처럼 계산되어진다.

$$Poff = Pcache\_acc + Mcache\_miss * (Pcache\_write + Ppad) , \quad (3)$$

여기서 *Pcache_acc* 은 캐쉬 블록을 접근할 때 소비되는 전력이며, *Mcache_miss* 은 캐쉬 접근 실패율이다. 또한 *Pcache_write* 은 캐쉬 접근 실패 시 캐쉬를 업데이트함으로써 소비되는 전력 소비이며, *Ppad* 은 온-칩 패드 슬롯에서 소비되는 소비 전력을 나타낸다. 마지막으로 *Ppad* 은 수식 (4)로 계산할 수 있다.

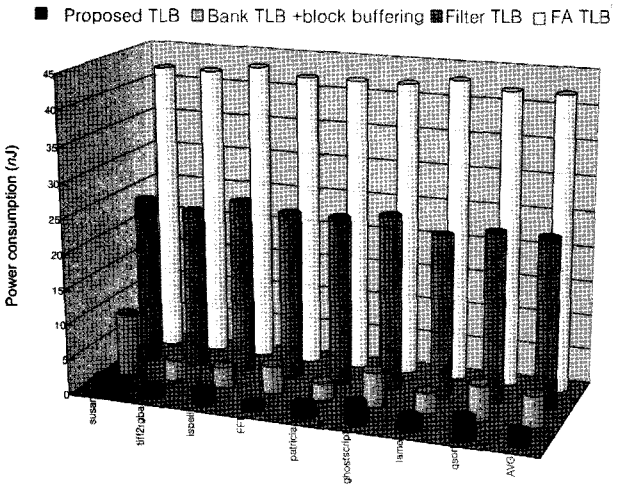
$$Ppad = 0.5 * Vdd2 * (0.5 * (Wdata + Waddr)) * 20pF , \quad (4)$$

여기서 *Wdata*와 *Waddr*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 2>처럼 요약되어진다.

(그림 7)은 제안된 TLB 구조와 다양한 TLB 구조의 소비 전력 차를 보여주고 있다. 제안된 연속적 접근 판별 TLB 구조의 경우, 추가적인 멀티플렉스, 비교기 등 가능한 대부분의 경우를 모두 고려하였다. 그림에서 알 수 있듯이 제안

〈표 2〉 시뮬레이션 변수 값

Mcache_miss	Pcache_acc	Pcache_write	Ppad	Poff
0.05	21.291 nJ	10.145 nJ	6.48 nJ	22.122 nJ



(그림 7) 제안된 구조와 다양한 TLB와의 소비 전력

된 구조의 경우 완전 연관 TLB에 비해 약 95% 정도 소비 전력 감소효과가 있으며, 필터 TLB에 비해 약 90%, 그리고 연속적 접근 판별 알고리즘을 사용하지 않은 동일한 구조에 비해서도 약 40%의 감소효과를 보이고 있다.

5. 결 론

이 논문의 주목적은 내장형 프로세서의 소비 전력을 줄이기 위하여 새로운 TLB 시스템을 설계하는 것이며 또한 구현이 간단하고 성능저하가 거의 없도록 하는데 중점을 두었다. 이러한 연구 목적을 달성하기 위하여 뱅크 구조와 블록 버퍼, 그리고 연속적 접근 판별 알고리즘을 제시하였다. 최근의 저전력 TLB 구조로는 필터 TLB 구조가 가장 각광을 받고 있지만 이러한 TLB 구조는 성능 저하라는 큰 단점을 내포하고 있다. 성능 저하를 막기 위하여 필터 버퍼의 엔트리 수를 확장할 수도 있지만 역시 소비 전력이 커지는 단점을 가진다. 뱅크 구조 역시 저전력 TLB 구조로써 좋은 구조라 할 수 있지만 소비 전력을 줄이는데 한계가 있는 구조이다. 이러한 단점들을 극복하고 성능 및 저전력에 효과적인 구조로써 최근에 참조되어진 데이터를 선택적으로 이용할 수 있는 새로운 TLB 구조를 제안하였다. 즉 기존의 계층 구조처럼 매번 모든 버퍼 엔트리들을 접근하는 것이 아니라 뱅크를 접근하는 시간을 이용하여 단지 두 비트만을 비교함으로써 해당하는 가상 페이지 번호가 버퍼에 존재하는지, 아니면 주 TLB에 존재하는지 판단함으로써 접근 시간을 줄이고 소비 전력 효과를 높이고자 하였다. 정확도가 가장 중요한 요소임으로 페이지의 특성과 많은 시뮬레이션의 수행을 통하여 최적의 비트를 찾고자 하였으며, 결과적으로 약 99%의 정확도를 보였다. 또한 전체 버퍼에서의 적중률이 90%이상으로, 주 TLB의 접근을 90%이상 막아줌으로써 소비 전력을 획기적으로 줄일 수 있었다. 시뮬레이션 결과에 따르면 소비 전력의 효과는 기존 완전 연관 구조에 비해 95%, 필터 구조에 비해 90%, 제안된 알고리즘을 사용하지 않은 동일 구조에 비해 40%의 소비 전력 감소 효과를 얻을 수 있었다.

참 고 문 헌

[1] Todd M. Austin and Gurindar S. Sohi, "Hign-bandwidth address translation for multiple-issue processors," In Proc. of the 32rd ACM Intl Symp. on Computer Architecture, pp. 158-167, May, 1996.

[2] T. Juan, T. Lang, and J. Navarro, "Reducing TLB Power Requirements," In Proc. of the International Symposium on Low Power Electronics and Design, 1997.

[3] I. Kadayif, A. Sivasubramaniam, M. Kandemir, G. Kandiraju, and G. Chen, "Generating Physical Addresses Directly for saving Instruction TLB Energy Efficiency," In Proc. of the International Symposium on Microarchitecture, 2002.

[4] S. Segars, "Low Power Design Techniques for Micro-processors," Tutorial Note of the ISSCC, Feb., 2000.

[5] M. B. Kamble and K. Ghose, "Energy-Efficiency of VLSI Cache: A Comparative Study," in Proc. of the 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," Int'l Symp. on Low Power Electronics and design, 1997.

[11] 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.

[12] ARM co., "ARM1136 Technical Reference Manual," http://www.arm.com/documentation/ARMProcessor_Cores/, 2003.

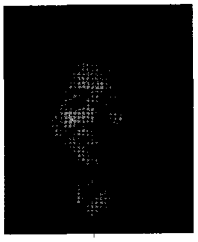
[13] K. Ghose and M. B. Kamble, "Reducing Power in Superscalar Processor Caches Using Subbanking, Multiple Line Buffers and Bit-Line Segmentation," Proc. International Symposium on Low Power Electronics and Design, pp.70-75, Aug., 199

[14] S. Manne, A. Klauser, D. Grunwald, F. Somenzi, "Low power TLB Design for High Performance Microprocessors," Univ. of Colorado Technical Report, 1997.

[15] Jan Edler and Mark D. Hill, "Dinero IV Trace-Driven Uniprocessor Cache Simulator," available from Univ. Wis., CS ftp site 1997.

[16] Glenn Reinman and Norm Jouppi, "An Integrated Cache Timing and Power Model," Compaq WRL Report, 1999.

[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

e-mail : leejh@gsnu.ac.kr

1999년 성균관대학교 제어계측공학과(학사)

2001년 연세대학교 컴퓨터과학과(석사)

2004년 연세대학교 컴퓨터과학과(박사)

2004년~현재 경상대학교 전기전자공학부 공학연구원 조교수

관심분야 : 지능형 메모리 시스템, 저전력/고성능 마이크로프로세서, 저전력 컴퓨팅, 내장형 시스템 및 SOC 시스템