

## 운영체제 도움 없이 멀티 페이지를 지원하는 저전력 TLB 구조

정 보 성\*, 이 정 훈\*

# Low Power TLB Supporting Multiple Page Sizes without Operation System

Bo-Sung Jung\*, Jung-Hoon Lee\*

### 요 약

비록 멀티 페이지 TLB는 성능을 향상시키는데 효과적이지만, 운영체제의 도움을 통한 기존의 방법은 사용자 응용 프로그램에서는 멀티 페이지를 사용할 수 없는 치명적인 단점을 가진다. 이에 본 논문에서는 운영체제의 지원 없이 멀티 페이지를 이용하여 고성능과 저전력을 얻을 수 있는 새로운 멀티 TLB 구조를 제안한다. 제안된 TLB는 작은 페이지를 위한 TLB와 큰 페이지를 위한 TLB로 구성되며, 모두 완전연관 뱅크 구조를 가지고 있다. 작은 페이지를 지원하는 S-TLB(Small TLB)는 큰 페이지를 지원하는 L-TLB(Large TLB)에서 추출된 작은 페이지를 저장하게 되며, L-TLB는 CPU로부터 요청된 작은 페이지를 포함한 큰 가상 페이지 주소를 저장하게 된다. CPU가 요청한 가상주소의 특별한 한 비트와 두 비트를 이용하여 S-TLB와 L-TLB의 각각의 하나의 뱅크만이 접근되며, 동시에 접근되는 엔트리 수 감소에 의해 에너지 소비를 줄일 수 있다. 또한 본 논문에서 효과적인 성능향상을 위해 간단한 1비트 LRU 정책을 제안하였다. 제안된 LRU 정책은 각 TLB 엔트리에 추가적인 1 비트를 사용하여 최근에 참조된 블록을 나타낸다. 이 방법은 간단하게 L-TLB로부터 가장 최근에 참조된 페이지를 선택할 수 있다. 시뮬레이션 결과에 따르면, 제안된 구조는 완전연관 사상 TLB, Dual TLB 그리고 ARM TLB에 비해 76%, 57%, 그리고 6%의 에너지\*지연시간을 줄일 수 있었다

▶ Keywords : TLB, 저전력, 메모리 관리 기법, 뱅크 구조, 멀티 페이지.

### Abstract

Even though the multiple pages TLB are effective in improving the performance, a conventional method with OS support cannot utilize multiple page sizes in user application. Thus, we propose a new multiple-TLB structure supporting multiple page sizes for high performance and low power consumption without any operating system support. The proposed TLB is organised as two parts of

•제1저자 : 정보성 •교신저자 : 이정훈

•투고일 : 2013. 6. 4, 심사일 : 2013. 8. 6, 게재확정일 : 2013. 11. 11.

\* 국립경상대학교 제어계측공학과(ERI, Dept. of Control & Instrumentation, Gyeongsang National University)

a S-TLB(Small TLB) with a small page size and a L-TLB(Large TLB) with a large page size. Both are designed as fully associative bank structures. The S-TLB stores small pages are evicted from the L-TLB, and the L-TLB stores large pages including a small page generated by the CPU. Each one bank module of S-TLB and L-TLB can be selectively accessed base on particular one and two bits of the virtual address generated from CPU, respectively. Energy savings are achieved by reducing the number of entries accessed at a time. Also, this paper proposed the simple 1-bit LRU policy to improve the performance. The proposed LRU policy can present recently referenced block by using an additional one bit of each entry on TLBs. This method can simply select a least recently used page from the L-TLB. According to the simulation results, the proposed TLB can reduce Energy \* Delay by about 76%, 57%, and 6% compared with a fully associative TLB, a ARM TLB, and a Dual TLB, respectively.

▶ Keywords : TLB, Low power, Memory management, Bank structure, Multiple page.

## I. 서 론

오늘날 컴퓨터 시스템내의 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 방법 중의 하나로써 제안되어왔다.

TLB는 프로세서 코어의 한 부분으로 L1 캐시 메모리와 병렬 접근 혹은 이전에 접근하게 된다[1]. 만약 필요로 하는 변환 정보가 TLB내에 존재 한다면, 시스템은 주어진 변환정보를 페이지 테이블에 접근 하지 않고 고속으로 요청된 물리 주소로 변환할 수 있다. 만약 TLB에 변환정보가 없다면, 페이지테이블을 참조하여 다시 참조하고 필요로 하는 정보를 TLB로 업데이트 시켜줘야 한다. 따라서 TLB의 접근 실패는 전체 시스템의 성능에 심각한 영향을 준다[2].

TLB는 매 클럭 접근이 이루어지므로, 일반적으로 고속의 접근 속도를 위해 완전연관 구조를 사용하고 있다. 비록 완전연관 구조가 좋은 접근 성공률을 보이지만, CAM (Content Addressable Memory)구성의 태그(Tag) 부분과 SRAM 구성의 데이터 부분들이 동적(dynamic) 회로로 구성되어 있기 때문에, 그들은 많은 에너지를 소비하는 큰 단점을 가진다. 이러한 TLB의 에너지 소비는 실제 프로세서 소비 에너지에 상당한 부분을 차지한다[3].

더욱이 메모리의 용량은 점점 증가 하고 있으며, 사용자 응용 프로그램 역시 작업 집합(Working Set)의 증가하고 있다. 결과적으로 TLB가 큰 작업 집합의 응용 프로그램의 효과적인 사용과 빠른 접근 시간을 가지기 어려워지고 있다[4].

AMD's Athlon, Intel's PentiumIII 그리고 여러 x86 프로세서는 기본적인 멀티 페이지 사용을 위해 두 페이지 크기의 TLB(separate TLBs)를 사용하고 있다. 이러한 방법은 완전연관 TLB에 대해 소비 에너지와 면적에 대한 단점을 피할 수 있다. 하지만, 서로 다른 페이지 크기를 사용하기 때문에 요청된 페이지가 있는 TLB를 예측 패턴을 최적화해야 한다. 만약 예측된 TLB의 접근 실패일 경우, 오히려 추가적인 에너지 소비와 접근 시간이 요구된다. 특히 TLB가 많은 블록을 가질 때 그 추가 비용은 더 늘어난다.

현재 멀티 페이지를 제공하는 TLB 시스템의 운용에 있어서 대표적인 방법은 커널 혹은 운영체제가 적합한 페이지 크기를 결정하여 할당하는 방법이다[5]. 하지만 이러한 운영체제가 멀티 페이지의 크기 할당 방법은 사용자 응용 프로그램에서 사용하기가 어려운 단점을 가진다.

따라서 TLB의 전체 성능향상을 위해 낮은 에너지 소비를 위한 구조적 방법 및 효과적인 페이지 관리를 위한 알고리즘이 필요하다.

본 논문에서는 운영체제의 도움 없이 하드웨어 관점에서 다양한 페이지 크기를 가지는 저전력 TLB의 구조 및 새로운 알고리즘을 제안한다. 제안된 TLB 시스템은 저전력을 위한 S-TLB와 멀티 페이지를 위한 L-TLB로 구성되며, 페이지

관리를 위한 알고리즘으로 최근 참조된 페이지 주소를 선택적으로 오랫동안 저장하므로, 멀티 페이지의 저장 가능한 페이지 주소를 효과적으로 관리하였다.

시뮬레이션 결과에 따르면, 에너지 소비와 성능을 고려한 에너지\*지연시간의 지표에서 제안된 TLB 구조는 완전연관 TLB에 비해 74%, Dual\_TLB에 비해 57%의 성능 향상을 이루었으며, 저전력 TLB인 ARM\_TLB와 비슷한 성능을 보였다.

본 논문의 나머지 부분은 다음과 같이 구성된다. II장에서 관련 연구가 소개되며, III장에서 제안된 TLB의 구조와 동작 원리에 대해 기술한다. 그리고 IV장에서는 성능 평가를 통한 시뮬레이션 결과를 비교하며, V장에서 결론을 맺는다.

## II. 관련 연구

기본적으로 TLB는 높은 접근 성공률을 위해 완전연관 구조로 이루어진다. TLB의 저전력 및 성능 향상을 위해 다양한 연구가 이루어지고 있다.

VS-TLB(6)는 기본적으로 Subblock TLB(7) 바탕으로 중간 및 큰 페이지에 대한 멀티 페이지 운용을 제안하였다. 비록, Subblock-TLB와 비교해서 접근 시간과 TLB 크기가 다소 증가할 지라도 중간 페이지 및 큰 페이지에 대한 효과적인 동작을 이루었다.

Fukunaga(8)는 거대 페이지를 사용하는 파일 시스템의 문제점인 메모리의 허비를 줄이기 위한 방법으로 사용된 메모리의 뒷부분의 작은 페이지 크기를 추가적으로 할당함으로써 거대 페이지의 단점을 보완하여 성능향상을 이루었다.

SpecTLB(9)는 직접적인 큰 페이지 저장 대신, 큰 페이지의 일부인 작은 페이지를 저장하게 된다. 작은 페이지의 많은 접근 실패시 페이지 테이블을 참조하지 않고, 예약된 큰 페이지에 대한 병렬적 접근으로 작은 페이지 주소 변환을 예측함으로써 TLB 접근 실패의 실행 대기 시간을 줄였다.

연구(6)(8)(9)은 멀티 페이지 운용에 있었어, 좋은 성능을 가지는 TLB 구조이지만, 이러한 멀티 페이지 운용은 운영체제에서 관리해야만 하는 단점이 있다.

Dual TLB(10)는 운영체제 도움없이, 하드웨어 관점에서 작은 2-페이지를 운용하므로 좋은 성능향상을 이루었다. 하지만, Dual TLB의 경우, 큰 페이지 저장을 위한 연속적인 작은 페이지가 존재 하지 않는다면, 큰 페이지를 위한 TLB의 운용이 극히 제한적이기 때문에 효과적으로 TLB의 블록 사용이 어렵다는 단점을 가진다.

Skewed Associative(SA)-TLB(11)는 멀티 인덱스 방

법으로 연관사상 구조에서 멀티 페이지를 지원함으로써 성능향상을 이루었다. 비록, SA-TLB가 멀티 페이지 지원 및 완전연관 구조에 비해 낮은 에너지 소비를 보인다 하여도, 완전연관 구조에 비해 낮은 접근 성공률과 멀티 페이지 검색을 위한 추가적인 접근 시간을 가지는 단점을 가진다.

ARM TLB(12)는 현재 효과적인 저전력 TLB 구조이다. 작은 크기의 완전연관 구조(Micro TLB)와 저전력을 위한 연관사상 구조(Main TLB)로 구성으로 효과적인 에너지 소비를 가진다. 하지만, 연관사상 구조로 인한 높은 접근 실패율과 Main TLB의 접근을 위한 추가적인 접근 시간을 가진다는 단점을 가진다.

연구(13)은 저전력 TLB를 위한 블록 버퍼링 기법과 뱅크 기법을 통합 방법으로 기존 TLB에 비해 좋은 성능향상을 이루었다. 하지만, 추가적인 희생 TLB 및 뱅크 TLB의 순차적인 접근으로 추가적인 접근이 요구되어진다.

연구(14)는 TLB의 성능향상을 위해 Prefetching 기법을 이용하였지만, 이 Prefetching 기법은 매 클럭마다 접근이 발생하는 TLB에 운용에 어려운 단점을 가진다.

연구(7)(10)(11)(12)는 하드웨어 관점에서 TLB의 성능향상을 가지는 구조이지만, 연구(7)(10)의 경우, 효과적인 TLB 블록 사용에 제한적이며, 연구(11)(12)의 경우, 저전력에 효과적인 구조지만, 낮은 접근성공률의 단점을 가지고 있다. 연구(13)은 추가적인 하드웨어 및 접근을 위한 추가적인 접근 시간을 가지는 단점을 보이고 있다.

따라서 TLB 성능향상을 위한 시스템 관점에서 효과적인 멀티 페이지 관리와 저전력을 위한 TLB구조 및 알고리즘이 필요하다. 제안된 TLB 구조는 S-TLB와 작은 블록 수를 가지는 L-TLB로 저전력에 효과적인 구조를 선택하였다. 시스템 관점의 멀티 페이지 운용을 위해 큰 페이지에 포함되는 작은 페이지를 S-TLB로부터 찾아 저장하므로 멀티 페이지를 효과적으로 사용할 수 있는 알고리즘을 제안하고자 한다.

## III. 시간지역성을 고려한 멀티 페이지 TLB 시스템

이 장에서는 연구 개발 동기에 대하여 설명하고 제안된 TLB의 구조와 메커니즘의 구체적인 동작에 대해서 설명한다.

### 3.1 제안된 TLB 시스템의 제안 동기 및 방법

이 논문의 주 목적은 운영체제 도움 없이, 하드웨어 관점에서 효과적인 멀티 페이지 관리를 위한 알고리즘 설계로 저전력 및 성능향상을 가지는 TLB 시스템을 구현하는 것에 그 목적을 두고 있다.

앞서 언급한 것과 같이, 일반적으로 TLB는 높은 접근 성공률을 위해 완전연관 구조로 이루어진다. 하지만, 완전연관 구조는 다른 메모리 구조들에 비해 높은 소비전력을 가진다는 큰 문제점을 가지고 있다. 또한 TLB의 성능향상을 위해 대부분 시스템들은 기본 페이지 외에 다른 페이지 크기를 가지는 멀티 페이지를 지원하고 있다. 이러한 멀티 페이지의 운용은 완전연관 구조의 높은 소비전력과 성능향상을 개선할 수 있다. 하지만, 이러한 멀티 페이지의 사용은 운영체제의 관리로 실제 구현이 어렵고 메모리 허비에 대한 문제점이 있다.

제안된 TLB 시스템은 작은 페이지 번호를 가지는 저전력을 위한 S-TLB(Small-TLB)와 다양한 크기의 페이지 번호를 위한 L-TLB (Large-TLB)로 구성되며, 모두 완전연관 구조로 이루어진다.

그림. 1은 제안된 TLB 구조를 나타낸 그림이다. S-TLB 들은 작은 가상 페이지 번호(virtual page number, VPN) 및 물리 페이지 번호(physical page number, PPN)를 가지며, 접근성공비트(H)를 가지고 있다. L-TLB는 큰 가상 페이지 번호를 가지므로, m개의 물리 페이지 번호를 가질 수 있다. 예로, S-TLB의 가상 페이지 번호 크기가 4Kbyte이고, L-TLB의 가상 페이지 번호 크기가 16Kbyte라면, L-TLB는 최대 4개의 물리 페이지 번호들을 저장할 수 있다. 또한 태그부분에 접근 성공 비트(H)와 물리 페이지 번호 영역에 참조비트(R)와 유효비트(V)로 구성된다.

성능향상을 위한 다양한 메모리 연구에서 알 수 있듯이, 프로그램 수행시 적합한 두 지역성(시간/공간 지역성)의 사용은 전체 시스템의 효과적인 성능향상을 가진다. 현재까지 TLB는 기본적으로 4Kbyte의 물리 페이지 주소를 저장하므로 충분한 공간 지역성을 보장 할 수 있다. 따라서 제안된 TLB구조는 공간 지역성 보다 시간 지역성을 효과적으로 사용을 위해 추가적인 비트를 가진다.

S-TLB와 L-TLB는 가상 페이지 번호가 저장된 블록의 접근성공을 나타내는 접근성공 비트(Hit bit)와 접근성공이 발생한 블록의 수를 나타내는 참조 블록 카운터 비트(Reference Block counter bits, R.B.C)를 가진다. 참조 블록 카운터는 각 TLB의 블록 수를 나타낼 수 있는 비트 수를 가진다. 예로 TLB가 8개의 블록을 가진다면, 블록 카운터

는 3비트를 가진다. 접근성공 비트는 현재 TLB의 어떤 블록에서 접근성공이 발생하면 1로 갱신되며, 이때 접근성공 비트가 갱신이 될 때, 참조블록 카운터가 증가하게 된다.

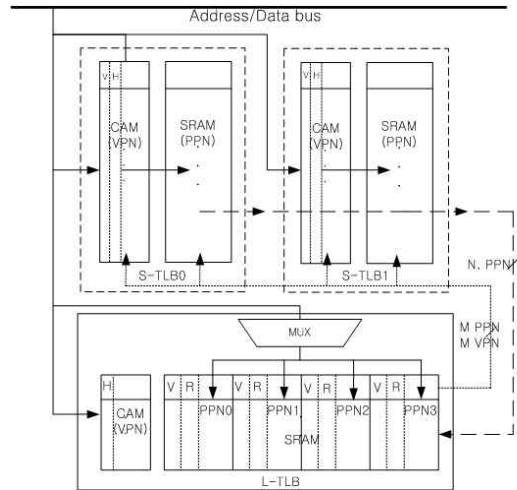


그림 1. 제안된 TLB 시스템 구조  
Fig. 1. The proposed TLB System architecture

만약, 참조블록카운터 모든 비트가 1일 경우, 접근성공이 발생한 TLB의 모든 접근성공 비트와 참조블록 카운터가 '0'으로 갱신 된다.

L-TLB는 m개의 S-TLB의 페이지 주소를 가질 수 있다. 앞서 언급한 것과 같이, 최근 사용한 페이지 주소를 가지는 것은 좋은 성능향상의 요소이다. 제안된 L-TLB 역시 최근 참조된 물리 페이지 번호를 선택하기위하여, L-TLB에 저장된 물리 페이지 번호당 참조 비트(Reference bit)를 가진다.

제안된 TLB 시스템은 S-TLB와 L-TLB가 동시에 접근이 이루어진다. S-TLB는 기존 완전연관 구조의 접근 동작과 동일하며, L-TLB는 m개의 물리 페이지 번호중 하나만 접근한다. 예로, 요청된 가상 메모리 번호의 태그 값이 110110이라면, 그리고 S-TLB가 2뱅크이면, 뱅크 선택은 최하위 비트인 '0'에 의해 선택된다. 그리고 L-TLB의 물리 페이지 번호 접근을 위해 최하위 2비트 값인 '10'에 의해 4개의 물리 페이지 번호 중 하나를 선택하게 된다.

TLB의 접근 실패시, 요청된 큰 가상 페이지 번호(예로 16Kbyte VPN)와 해당 물리 페이지 번호 (예로 4Kbyte PPN)는 L-TLB에 저장되며, 만약 S-TLB에 요청된 큰 가상 페이지 번호에 포함되는 나머지 물리 페이지 번호가 존재한다면, 그 정보들 역시 L-TLB에 저장되며, 선택되어진 S-TLB

의 블록들은 무효블록으로 처리된다. 자세한 동작은 다음 절에서 자세히 설명되어 진다.

### 3.2 구조적 특징과 동작모델

제안된 TLB 동작 원리는 다음과 같다. 먼저 가상 주소가 중앙 처리 장치로부터 발생하면, 요청된 VPN의 최하의 1비트에 의한 S-TLB와 최하의 2비트에 의해 L-TLB의 작은 페이지들 중 하나의 작은 페이지에 동시 접근이 이루어진다.

그림 2는 제안된 TLB의 기본적인 접근성공 및 접근 실패 일 때 동작을 구체적으로 나타낸 그림이다. S-TLB와 L-TLB는 4개의 블록들을 가지며, L-TLB는 4개의 PPN를 가질 수 있다고 가정 하였다. L-TLB의 VPN는 대문자 알파벳(A, B, C ...)으로 나타내며, S-TLB의 VPN은 첨자를 가지는 소문자 알파벳(a1, a2, a3, a4, b1, b2 ...)으로 나타내고 있다. 따라서 L-TLB의 가상 페이지 번호(A)는 4개의 S-TLB 가상 페이지 번호(a1, a2, a3, a4)를 가지고 있다.

#### 1) TLB 접근성공의 경우 :

CPU로부터 요청된 VPN이 S-TLB 혹은 L-TLB에 존재한다면, 접근성공이 발생한 TLB의 접근성공 비트가 '1'로 갱신되며, 이때, 참조블록 카운터 역시 갱신이 된다. 추가적으로 L-TLB의 경우, 접근성공이 일어난 해당 PPN의 참조비트가 '1'로 갱신 일어난다. 만약 해당 TLB의 모든 참조 블록 카운터가 '1'이면, 해당 TLB의 접근성공 비트와 참조 블록 카운터가 모두 '0'으로 갱신되어진다.

그림 2(a)는 현재 S-TLB 및 L-TLB의 상태를 나타내고 있으며, 그림 2(b)는 가상 페이지 번호 b0와 h2의 순차적인 요청시 제안된 TLB의 접근성공 동작을 나타낸 그림이다. VPN b0는 S-TLB0에서 접근성공이 발생하며, 이때 b0의 접근 성공비트는 '1'로 갱신되어야 한다. 이때 S-TLB0의 모든 H 비트가 모두 1이기 때문에 참조블록 카운터는 00으로 바뀌게 되고 TLB0의 접근 성공 비트도 모두 0으로 바뀌게 된다.

VPN h2의 접근 성공은 L-TLB (VPN:H)에서 발생하게 된다. 이전상태(그림 2(a))에서 h2를 포함하는 큰 가상 페이지 번호인 H 저장된 블록은 접근성공 비트가 '1'이므로, 단지 h2가 저장되어 있는 물리 페이지 번호가 저장된 곳의 참조비트(R)만 '1'로 갱신이 된다.

#### 2) TLB 접근 실패:

##### i) L-TLB에 무효블록이 존재할 경우.

L-TLB에 무효 블록이 존재할 경우, S-TLB에는 어떠한 페이지 정보도 존재하지 않기 때문에 아무런 동작을 수행하지

않는다.

제안된 TLB는 큰 VPN이 존재할 경우와 그렇지 못할 경우만 존재한다. 요청된 큰 VPN이 L-TLB에 없을 경우 TLB 접근 실패가 발생하게 되고, 큰 VPN을 멀티 TLB의 무효블록에, 요청한 PPN을 페이지 테이블로부터 L-TLB의 해당 위치에 저장하며 유효비트(V)만 '1'로 갱신된다.

만약 요청된 큰 VPN이 L-TLB의 태그에서는 접근 성공이지만 해당 PPN의 유효 비트가 '0'인 경우에는 TLB 접근 실패가 된다. 이 경우에는 해당 PPN을 페이지 테이블로부터 L-TLB의 해당 위치에 저장하며 유효비트(V)를 '1'로 갱신하게 된다.

##### ii) L-TLB가 모두 유효블록 일 경우.

TLB의 접근 실패시 새로운 VPN을 멀티TLB에 저장하기 위해서는 L-TLB의 참조블록 카운터가 '00'이 아니면, 접근성공 비트(H)가 '0'인 블록을 희생블록으로 선택하며, 아니면, FIFO 알고리즘으로 희생블록이 선택되어진다. 이때 방출되는 희생 블록의 유효한 물리 페이지 번호는 해당 S-TLB에 작은 VPN(예로 4Kbyte)과 함께 저장되어진다. 이때, S-TLB에 저장되어지는 페이지 번호의 참조비트(R)가 '1'이라면, 이 참조비트의 정보는 S-TLB의 해당 블록의 접근성공 비트(H)에 동일하게 '1'로써 갱신됨과 동시에 해당 S-TLB의 참조블록 카운터 역시 새로운 값으로 갱신된다. 결론적으로 S-TLB의 접근 성공 비트 수가 1이면 참조블록 카운터는 '01'이며, 접근성공 비트 수가 4이면 참조블록 카운터는 '00'로 갱신됨과 동시에 모든 접근성공 비트가 '0'으로 갱신된다. 만약, L-TLB에서 S-TLB로 페이지 정보가 갱신될 때 S-TLB에 유효 블록이 없으면 해당 페이지 수만큼에 해당하는 페이지 정보가 희생 블록으로 선택되어야 한다. 만약, 참조블록 카운터가 '00'일 경우 FIFO 알고리즘에 의해 희생 블록은 선택되어지며, 아니면 접근성공 비트(H)가 '0'인 블록을 희생블록으로 선택되어진다.

새로운 큰 VPN이 L-TLB에 저장된 후, S-TLB로부터 큰 VPN에 속한 작은 VPN들이 존재한다면, 그에 해당하는 PPN들은 L-TLB에 저장되며, 물리 페이지 번호의 유효비트(V)만이 갱신되어진다. 이때 L-TLB에 저장된 모든 PPN의 참조비트(R)는 모두 '0'으로 갱신이 이루어진다. 또한, L-TLB에 저장되어진 S-TLB의 작은 VPN의 블록은 무효블록을 처리되며, 만약 무효블록 처리된 S-TLB의 접근성공 비트(H)가 '1'이었다면, 그 S-TLB의 참조블록 카운터 역시 다시 갱신이 된다. 즉, S-TLB의 참조블록 카운터가 '11'이며, L-TLB에 저장되는 작은 VPN 블록의 접근성공 비트(H)가

1이라면, 그 블록은 무효 블록으로 처리 후 참조블록 카운터 '10'으로 갱신이 된다.

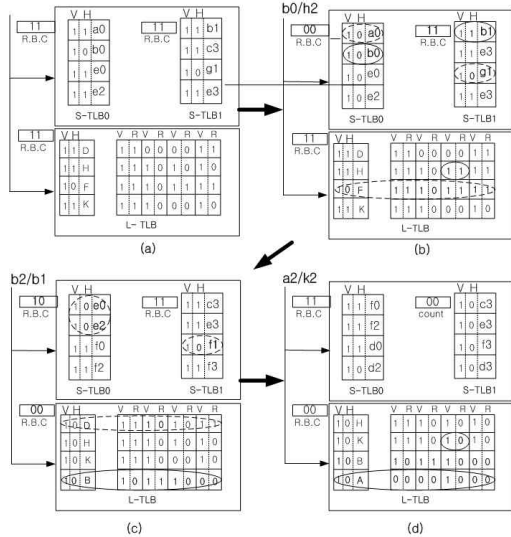


그림 2. 제안된 TLB 시스템의 동작 모델  
Fig. 2. Operation model of the proposed TLB System

그림 2(c)(d)는 제안된 TLB 접근 실패일 때 동작을 나타낸 그림이다. 그림에서 희생 블록은 점선으로 표시되며, 페이지 정보 갱신이 발생하는 블록은 실선으로 표시되어져 있다. 제안된 TLB의 그림 2(c)의 이전상태(그림 2(b))에서 VPN b2, b1이 순차적으로 요청된다며, 먼저, VPN b2는 TLB 접근 실패가 발생하며, 그림 2(b)에서 L-TLB의 희생블록은 접근 성공 비트가 '0'인 큰 VPN F의 블록이 선택되며, 유효 PPN이 저장된 VPN f0, ~ f3이 각 S-TLB에 저장된다. 그리고 L-TLB에 큰 VPN B, PPN b2가 S-TLB로부터 VPN b0와 b1과 함께 저장된다. 이때 S-TLB로부터 VPN b1, b2는 접근성공 비트(H)가 1이므로, 각 S-TLB의 참조 블록 카운터가 00과 10으로 갱신이 된다. L-TLB에 저장된 VPN b0, b1, b2의 PPN의 참조비트(R)는 '0'의 값을 가지며, 큰 VPN B의 블록 접근성공 비트 역시 '0'의 값을 가진다. VPN f0~f3이 S-TLB에 저장될 때, S-TLB역시, 모든 블록이 유효블록으로 접근성공 비트(H)가 0인 VPN a0, g1의 블록이 희생블록으로 선택된다. S-TLB에 저장되어지는 VPN f0~f3중 f0, f2 그리고 f3의 PPN의 참조비트(R)가 '1'이므로, S-TLB0의 참조블록 카운터가 다시 00에서 10으로 갱신되며, S-TLB1의 참조블록 카운터는 10에서 11로 갱신된다. VPN b1이 요청시, L-TLB에서 접근성공이 발생하고,

VPN B가 저장된 블록의 접근성공 비트(H)와 PPN의 참조비트(R)가 '1'로 갱신됨에 따라, 참조블록 카운터가 11에서 00로 갱신이 된다.

마지막으로, 그림 2(c)에서 VPN a2, k2를 요청시, 먼저 a2에 의해 TLB의 접근실패가 발생하고, L-TLB에서 FIFO 알고리즘에 의해 큰 VPN D의 블록이 희생 블록으로 선택된다. 이때 유효 물리 페이지 번호를 가지는 VPN d0~d3이 S-TLB에 저장된다. S-TLB0는 VPN e0와 e2가 존재하는 블록이 희생 블록으로 선택되며, d0의 참조비트(R)로 인해 S-TLB0의 참조블록 카운터가 '11'로 갱신된다. 그리고 S-TLB1의 경우, 희생블록은 가상 페이지 번호 f1이 존재하는 블록이 선택되며, d1이 저장된다. 그리고 d3이 S-TLB1에 저장될 때, 비록 d1이 최근에 저장된 VPN이지만, 접근성공 비트(H)가 '0'이므로 d1이 e3, cf 그리고 f3보다 먼저 희생블록으로 선택되고 d3이 저장되며, 참조블록 카운터가 '00'으로, 모든 접근성공 비트(H)가 '0'으로 갱신된다.

k2의 요청시, L-TLB에 큰 VPN K가 존재하지만, VPN k2의 PPN이 존재하지 않으므로 TLB접근 실패가 발생하며, 단지, k2에 속하는 PPN을 저장하고 유효비트(V)만 1로 갱신된다.

#### IV. 시뮬레이션을 통한 성능평가

시뮬레이션 환경과 성능 평가 지표, 그리고 소비전력에 대한 다양한 시뮬레이션의 결과가 소개되어 있다. 제안된 TLB 시스템의 성능평가 지표로 평균 메모리 접근 시간(Average Memory Access Time), 소비 전력(Energy Consumption) 그리고 에너지 \* 지연 시간 곱(Energy\* delay product)을 사용하였다. 평균 메모리 접근 시간을 위해 Simplescalar-3.0[15] 과 에너지 소비를 위해 CACTI-3.0 [16]을 수정하였다. 성능평가를 위한 벤치마크로 본 논문에서는 Spec cpu2006[17]를 선택하였다. <표 1>은 시뮬레이터를 위한 기본 변수 값이다.

표 1. 시뮬레이션 변수들  
table. 1. Simulation Parameter.

변수들	값
CPU clock	3.6Ghz
Memory clock	266Mhz
Memory bandwidth	1.6Gytes/sec
Memory latency	120ns

제안된 TLB 시스템의 성능 평가를 위한 비교 TLB구조로

본 논문에서는 가장 보편적인 완전연관 TLB 구조(fully associative TLB), 저전력에 효과적인 ARM TLB 구조, 그리고 운영체제 도움없이 2 페이지 구조를 가지는 이중 TLB 구조(Dual TLB)를 선택하였다.

제안된 TLB 시스템과 성능비교를 위해 완전연관 TLB는 64개의 블록을 가지며(FA\_64), ARM TLB는 Micro TLB는 8개의 블록을 Main\_TLB의 64개의 블록을 가진다(ARM\_8,64). 그리고 이중 TLB는 32개의 블록을 가지는 작은 페이지 TLB와 8개의 블록을 가지는 큰 페이지 TLB(Dual\_32,8)로 구성된다. 그리고 제안된 TLB는 작은 페이지 블록을 가지는 S-TLB는 각 8개의 블록, L-TLB는 8개의 블록을 가진다. 모든 비교 TLB 및 제안된 TLB는 4Kbyte 페이지 크기에 대한 변환정보를 가진다.

그림 3은 평균 메모리 접근 시간을 나타낸 그림이다. 결과적으로 제안된 TLB 시스템은 ARM\_TLB에 비해 25%의 성능향상을 보였으며, 완전연관 TLB와 이중 TLB에 비해 각각 단지 8%, 6%의 성능저하만 보였다. ARM\_TLB의 경우, 저전력을 위해 Main\_TLB가 2-웨이 연관구조로 구성되어 완전연관 구조에 비해 높은 접근 실패율을 가진다. 또한 Micro\_TLB와 Main\_TLB가 순차적인 접근으로 Main\_TLB의 접근시 추가적인 접근 시간이 요구된다. 그림에서 알 수 있듯이, 제안된 TLB 시스템은 bzip2, gombk에서 완전연관 TLB와 이중 TLB와 성능 차이를 보이고 있다. 관련 수행 연구에서 다양한 TLB 크기의 시뮬레이션 결과 특정 블록의 수에서 좋은 성능향상을 보였다. 제안된 TLB 시스템은 bzip2, gombk의 작업 세트(working set) 불리한 TLB 블록수를 가지고 있기 때문이다.

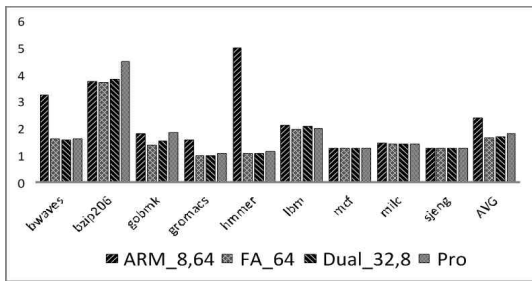


그림 3. 평균 메모리 접근 시간(AMAT, cycle)  
Fig. 3. Average Memory Access Time(AMAT, cycle)

그림 4는 제안된 TLB와 비교 TLB의 에너지 소비를 나타낸 그림이다. ARM\_TLB는 그림 3의 평균 메모리 접근시간에서 가장 높은 성능저하를 보이는 반면, 에너지 소비에서는

가장 좋은 성능 향상을 가진다. 그리고 완전연관 TLB는 가장 높은 에너지 소비를 보이고 있다. 실제, 완전연관 구조에서 CAM구동은 에너지 소비의 많은 부분을 차지하고 있다. 특히 블록의 수에 비례하여 에너지 소비가 증가한다. 따라서 완전연관 TLB는 많은 블록 수로 인해 높은 에너지 소비를 가지며, 비록 이중 TLB가 큰 페이지 TLB가 적은 블록 수를 가진다하여도, 작은 페이지 TLB에서 많은 블록 수를 가지므로 높은 에너지 소비를 보여주고 있다. 반면, ARM\_TLB의 경우, Micro\_TLB는 적은 블록의 수와 2웨이 구조의 Main\_TLB로 아주 낮은 소비 전력을 가진다.

제안된 TLB 시스템은 S-TLB와 L-TLB 모두 적은 수의 블록을 가지므로 낮은 에너지 소비에 효과적이다.

에너지 소비 결과 제안된 TLB는 완전연관 TLB에 비해 76%, 이중 TLB에 비해 60%의 성능향상을 이루었다. 반면, ARM\_TLB에 비해 30%의 높은 에너지 소비를 가진다. 하지만, ARM\_TLB의 경우, 제안된 TLB 구조에 비해 많은 TLB 블록을 가지고 있다.

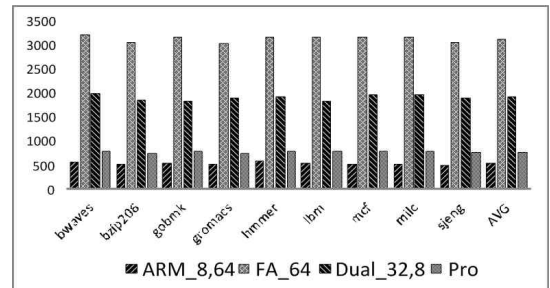


그림 4. 에너지 소비(uJ)  
Fig. 4. Energy consumption(uJ)

그림 5는 에너지\*지연시간에 대한 그림이다. 에너지 \* 지연시간 성능지표는 서로 다른 구조에 대해 성능평가를 위한 효과적인 지표이다. 제안된 TLB 시스템은 완전연관 TLB와 이중 TLB에 비해 74%, 57%의 성능 향상을 보인다. 하지만, ARM\_TLB에 대해서는 약 8%의 성능 저하를 가진다. 비록, ARM\_TLB가 높은 평균 메모리 접근 시간을 가지지만, 다른 비교 TLB에 비해 아주 낮은 에너지 소비를 가진다. 따라서 에너지 \* 지연시간 지표에서 좋은 성능을 보인다. 하지만, 제안된 TLB 구조는 에너지 소비가 높은 완전연관 구조임에도 불구하고, 단지 ARM\_TLB에 비해 8%의 성능저하만을 가지고 있다.

하지만, ARM\_TLB의 경우, 제안된 TLB 구조에 비해 많은 용량을 사용하고 있다.

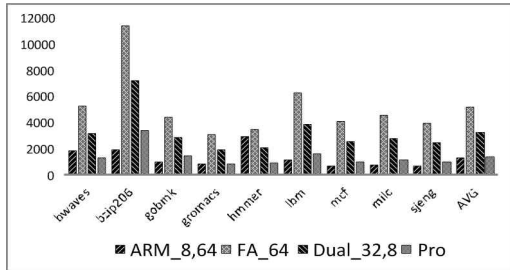


그림 5. 소비 에너지 \* 지연시간  
Fig. 5. Energy \* delay product

### V. 결론

본 논문에서는 저전력을 위한 TLB 구조 및 효과적인 페이지 관리를 위한 알고리즘을 제안하였다. 제안된 구조는 작은 페이지를 가지는 S-TLB와 다중 페이지를 위한 L-TLB로 모두 완전연관 구조로 이루어진다. 뱅크 구조 및 subblock 구조는 이미 메모리 연구에서 효과적인 에너지 소비의 한 방법이다.

또한, 제안된 TLB 시스템은 초근 참조된 가상 페이지 번호를 TLB에 오래 저장하므로, TLB 접근 실패시, 큰 가상 페이지 번호에 포함되는 작은 가상 페이지 번호를 효과적으로 가질 수 있는 방법이라 할 수 있다. 시뮬레이션 결과, 이러한 알고리즘 동작은 제안된 TLB 시스템에서 S-TLB로부터 L-TLB에 다양한 크기의 페이지 정보를 제공할 수 있었다.

제안된 TLB는 시뮬레이션 결과 접근성공률이 좋은 완전연관과 버퍼에 비해 적은 용량으로 비슷한 평균메모리 접근 시간을 가질 수 있었다.

제안된 TLB구조는 보편적으로 사용되는 단일 완전연관 TLB 구조에 비해 에너지\*지연시간에 대해 74%의 성능 향상을 보이며, 시스템 관점의 이중 페이지를 가지는 Dual TLB에 대해 57%의 성능향상을 이루었다. 저전력에 효과적인 ARM TLB에 대해 약 8%의 성능저하를 보이지만, ARM\_TLB의 경우, 제안된 TLB 구조에 비해 많은 용량을 사용하고 있다. TLB를 위한 용량 대 성능향상을 고려시, 제안된 TLB 구조는 ARM\_TLB에 비해 좋은 성능을 향상을 가진다 할 수 있다.

따라서 제안된 TLB 구조는 보편적으로 사용되는 완전연관 TLB, 시스템 관점의 이중 페이지를 지원하는 Dual TLB, 그리고 저전력에 효과적인 ARM\_TLB에 비해 작은 용량의 효과적인 저전력 TLB라 할 수 있다.

### 참고문헌

- [1] X. G. Qiu and M. Dubois, "Moving Address Translation Closer to Memory in Distributed Shared-Memory Multiprocessors," IEEE Tran. on Parallel and Distributed Systems, Vol. 16, No 7, pp.612-623, Mar. 2005.
- [2] T. W. Barr, "Exploiting Address Space Continuity to Accelerate TLB Miss Handling," Master degree paper of Rice University, 2010.
- [3] A. Basu, M. D. Hill, and M. M. Swift, "Reducing Memory Reference Energy with Opportunistic Virtual Caching," In Proceedings of International Symposium on Computer Architecture, pp.297-308, 2012.
- [4] R. Bhargava et al., "Accelerating Two-Dimensional Page Walks for Virtualized Systems," Proceedings of the 13th international conference on Architecture support for programming languages and operation system, pp.26-35, 2008.
- [5] B. Pham, V. Vaidyanathan, A. Jaleel and A. Bhattacharjee, "CoLT: Coalesces Large-Reach TLBs," Annual IEEE/ACM International Symposium on MICRO, pp.258-269, Dec. 2012.
- [6] C. H. Pack, D. Y. Pack, "Increasing TLB Reach with Multiple Pages Size Subblocks," 21st IEEE International Performance, Computing and Communications Conference, pp.123-130, 2002.
- [7] M. Talluri and M. D. Hill, "Surpassing the TLB performance of superpages with less operating system support," in Proc. of the 6th Symposium on Architectural Support for Programming Languages and Operating systems, pp.171-182, Oct. 1994.
- [8] T. Fukunaga and T. Sueyoshi, "Improvement of parallel processing performance by using two kinds of Huge Page," Automation and Systems International Conference on Control, pp.2662-2666, Oct. 2008.
- [9] T. W. Barr, A. L. Cox, and S. Rixner, "SpecTLB:



a mechanism for speculative address translation,” In Proceeding of the 38th annual international symposium on Computer architecture, pp.307-318, 2011.

- [10] J. H. Lee and S. D. Kim, “A dynamic TLB management structure to support different page sizes,” Proceedings of the Second IEEE Asia Pacific Conference on ASICs, pp.299-302, Aug. 2000.
- [11] A. Seznec, “ Concurrent support of Multiple page sizes on a skewed associative TLB,” IEEE transactions on computers, Vol. 53, pp.924-927, July, 2004.
- [12] cortex-A9: technical reference manual, 2008.
- [13] Y. J. Chang, “Two New Techniques Integrated for Energy-Efficient TLB Design,” IEEE Transactions on Very Large Scale Integration System, Vol. 15, No. 1, Jan. 2007.
- [14] A. Bhattacharjee and M. Martonosi, “Inter-Core Cooperative TLB Prefetchers for Chip Multiprocessors,” Proceedings of the 15th edition of Architecture support for programming languages and operation system, pp.359-370, 2010.
- [15] D. Burger and T. M. Austin, “The SimpleScalar tool set, version 2.0, Technical Report TR-97-1342,” University of Wisconsin-Madison, 1997.
- [16] G. Reinman. and N. P. Jouppi, “CACTI 3.0: An integrated cache timing and power, and area model,” Compaq WRL Report, Aug. 2001.
- [17] SPEC Benchmark Suite. <http://www.spec.org>

## 저 자 소 개



### 정 보 성

2008년 2월: 경상대학교 제어  
계측공학과(석사)  
2008년~현재: 경상대학교 제어  
계측공학과 박사과정  
관심분야: 마이크로프로세서, 캐쉬  
및 플래시 메모리  
Email : blueking80@gnu.ac.kr



### 이 정 훈

2004년 2월: 연세대학교 컴퓨터  
과학과(박사)  
2004년~현재: 경상대학교 제어계측  
공학과 부교수  
관심분야: 고성능 컴퓨팅, 내장형  
시스템 및 SOC 시스템  
Email : leejh@gnu.ac.kr