저전력 온칩 메모리에 관한 연구 동향 및 개발 방향

연세대학교 이정훈* · 최진혁 · 김신덕**

1. 서 론

정보화 사회가 빠르게 세분화되고 거대화됨에 따라 다양한 형태의 정보화 기기의 사용이 급속히 확대되고 있다. 특히 이동 통신 단말기, MP3 재생기, 포터블 컴퓨터, 셋톱 박스, PDA와 같은 다양한 포터블 시스템의 출현과 보급은 개인용 컴퓨터 등에 사용되어 온 범용 마이크로 프로세서뿐만 아니라 고성능 내장형 프로세서에 대한 요구를 증대시키고 있다. 주로 제어용으로 사용되던 기존의 내장형 프로세서와는 달리 이러한 고성능 내장형 프로세서는 기존의 마이크로 프로세서의 고속 컴퓨팅 능력과 함께 높은 메모리 대역폭, 효과적인 메모리 계층 구조, 메모리 운용 유닛을 통한 가상 메모리 지원 등의 기능을 기본적으로 요구하고 있다. 뿐만 아니라 포터블 시스템의 배터리 수명을 가능한 연장시키려는 소비자들의 요구를 만족시켜주기 위해서는 프로세서의 기능과 성능을 향상시키는 것뿐만 아니라 전력 소모를 줄이는 것이 무엇보다도 중요하다.

일반적으로 캐쉬나 TLB(Translation lookaside buffer)와 같은 프로세서의 온칩 메모리에서 소모되는 전력은 전체 칩에서 소모되는 전력의 상당부분을 차지한다. 예를 들어, 대표적인 RISC 내장형 프로세서인 StrongARM(2.0Vdd, 0.35μ, 233 MHz, 1W typical) [1]의 전력 소모를 살펴보면, 명령어 캐쉬, 데이터 캐쉬, TLB의 전력 소모가 칩 전체 전력 소비의 27%, 17%, 17%를 차지하는 것을 알 수 있다. 이렇게 온칩 메모리가 많은 전력 소모를 보이는 이유는 다음과 같다. 첫째, 온칩 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 CAM(Content addressable memory)은 내부 비교 로직과 부가적인 매치 라인(match line)들로 인해 SRAM보다도 훨씬 많은 전력을 소비한다. 둘째, 온칩 메모리 시스템은 매우 자주 접근되는 경향이 있다. 특히, 명령어의 경우 TLB와 캐쉬 메모리는 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 온칩 메모리 시스템을 접근하거나 오프칩 메모리 접근을 위해서 I/O 패드를 구동해야 한다. 일반적으로 I/O 패드의 정전용량(capacitance)은 온 칩 정전 용량 보다 훨씬 크다. 따라서 온칩 메모리 접근 실패 횟수를 줄이는 것이 저전력 메모리 시스템을 설계하기 위한 가장 기본 접근 방법이 된다.

일반적으로 온칩 메모리 시스템의 저전력에 대한 고려는 설계 과정의 각 단계에서 이루어 질 수 있으며 이는 상위 수준의 알고리즘 선택, 시스템 집적, 아키텍처 설계에서부터 하위 레벨의 게이트/회로 설계, 공정 단계를 포함한다. 이러한 다양한 저전력 설계 단계 중 아키텍쳐, 알고리즘 및 시스템 수준에서의 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 포괄적인 개념을 다루기 때문에 가장 중요한 비중을 차지한다. 본 논문에서는 저전력 온칩 메모리의 구현을 위해 기 제안된 주요 아키텍쳐 및 시스템 수준에서의 설계 기법들을 소개하고 기존 기법들의 단점들을 극복하는 향상된 시스템 수준의 저전력 메모리 설계 기법을 제안한다.

* 학생회원
** 정 회 원

2. 저전력 온칩 메모리 설계 방향

온칩 메모리는 크게 캐쉬 메모리와 주소 변환 버퍼(Translation lookaside buffer: TLB)로 구성되어진다. 캐쉬 메모리는 컴퓨터 시스템의 성능 향상에 가장 큰 병목으로 지적되고 있는 프로세서와 메모리의 성능 격차에 의해 발생하는 성능 저하를 완화시키기 위해 사용되는 온칩 메모리이다. 캐쉬의 성능을 향상시키는 가장 효과적인 방법은 프로그램 수행 특성에 내재되어 있는 시간적 지역성과(temporal locality) 공간적 지역성(spatial locality)을 활용하는 것이다. 시간적 지역성은 최근에 참조되어진 블록이 가까운 시간 내에 다시 참조될 확률이 대단히 높다는 것을 의미하며, 공간적 지역성이란 참조가 일어난 블록의 이웃 블록이 참조될 확률이 대단히 높다는 것을 의미한다[2]. 그러나 이러한 두 가지 지역성은 캐쉬 크기가 고정되어 있을 경우 서로 상반되는 특성을 가지고 있다[3]. 즉 블록의 크기가 커질수록 공간적 지역성은 효과적으로 반영할 수 있지만 캐쉬 엔트리 수의 감소로 시간적 지역성에 대한 효과는 감소된다. 이러한 상반된 특성을 가진 지역성을 효과적으로 이용하여 고성능 캐쉬 메모리를 설계하고자 한 연구가 활발히 진행되어 왔다. 이와 더불어 오늘날 내장형 프로세서는 소비 전력에 대한 비중이 점차 커짐에 따라 저전력에 주안점을 두고 연구가 활발히 진행중이다.

주소 변환 버퍼(TLB)는 가상 주소(virtual addresses)를 물리 주소(physical addresses)로 변환하기 위한 페이지 테이블(page table)을 구성하는 캐쉬 메모리이다[4]. 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 성능 향상을 위한 전형적인 방법은 크게 세 가지 방식으로 구분되어진다. 첫째, 많은 엔트리 수(entry number)를 지원하는 방법, 둘째 페이지 크기(page size)를 증대시키는 방법[5], 셋째, 다중 페이지 크기 (multiple page size)를 지원하는 방법이다[6]. 그러나 TLB 엔트리 수가 증대되어 진다면, 참조 시간 지연(latency)이라는 역효과가 나타나며, 또한 일반적으로 TLB는 CAM으로 구현되어지기 때문에 참조 시마다 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 프로세스의 수가 제한을 받게 된다는 단점도 가지게 된다. 그러므로 현재 TLB 성능을 높이기 위한 가장 효과적인 방법은 다중 페이지 크기를 지원하는 방법이다. 다중 페이지 크기를 지원하는 방법들 중에서 가장 적합한 것은 운영체제나 컴파일러로부터 일정한 정보를 받아 가장 적합한 페이지 크기를 TLB에 할당하는 것이다. 그러나 운영체제의 시스템(kernel) 영역에서는 이러한 방식이 가능하나 사용자(user) 영역에서는 현실적으로 이러한 방식을 지원하기 어려운 단점을 가진다.

온칩 메모리의 접근 실패 횟수를 줄이는 것은 바로 저전력 메모리 시스템을 설계하기 위한 가장 기본 접근 방법이 될 수 있다. 이외에도 저전력 온칩 메모리를 위해 회로 레벨 및 공급 전압을 이용하는 방법이 제시되어졌다. 즉, CAM 메모리 셀 자체를 변형시키는 방법[7], 낮은 공급 전압을 이용하는 방법[8], 그리고 전압 스케일링을 이용한 방법들이 있다[9]. 그러나 메모리 셀 자체를 변화시키는 것은 하드웨어 비용이 증가하는 단점을 가지게 되고 또한 낮은 공급 전압을 제공하는 방법은 다른 기술적인 문제를 해결해야하는 어려운 작업이라 할 수 있다. 또한 전압 스케일링은 컴파일러 수정및 성능 저하를 초래할 수 있는 단점들이 있다. 그럼으로 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 기법 중 가장 포괄적인 개념을 다루기 때문에 대단히 중요한 비중을 차지한다고 볼 수 있다.

3. 기존의 시스템 수준의 저전력 온칩 메모리 설계 기법

대표적인 시스템 레벨의 저전력 온칩 메모리 설계 기법은 블록 버퍼링 메커니즘과 뱅크 메커니즘, 그리고 필터 메커니즘을 이용하는 것이다. 예시된 모든 설명 및 그림은 저전력 설계 기법 중 캐쉬에 비해 상대적으로 연구가 미약한 TLB을 기준으로 설명한다.

3.1 블록 버퍼링 메커니즘(block buffering)

블록 버퍼링 메커니즘은 시간적 지역성을 최대한 이용하여 소비 전력을 줄이고자 한 연구로써 캐쉬 또는 TLB 모두 이용 가능한 기술이다[10, 11, 12]. 필터 메커니즘과 유사한 기술이지만 다수의 버퍼 메모리 사용 대신 하나의 태그 버퍼만을 이용하고 데이터 출력 드라이브(data output driver or latch)를 이용하여 추가적인 하드웨어 비용을 최소화한 기술이다.

하나의 주소가 생성되어지면 주 온칩 메모리를 참조하기 전 바로 앞에서 생성된 주소가 저장되어 있는 태그 버퍼와 비교 수행을 하게된다. 만약 태그 버퍼에서 접근 성공이 일어나면 이는 앞선 주소에서 이미 인출되어진 데이터가 출력 드라이버에 그대로 남아 있기 때문에 다시 태그 메모리나 데이터 메모리의 접근 없이 데이터 출력 드라이브에 존재하고 있는 데이터를 그대로 이용 할 수 있다. 그럼으로 주 온칩 메모리의 접근 소비 전력을 줄일 수 있다. 그러나 블록 버퍼의 태그 부분에서 접근 실패가 발생하면 이는 같은 주소가 연속적으로 생성되지 않았음을 나타냄으로 정상적으로 주 온칩 메모리 접근을 수행하게 된다.

이 기술의 가장 큰 단점은 한 사이클 내에 블록 버퍼의 적중/실패를 결정함과 동시에 접근 실패의 경우 주 온칩 메모리의 접근 적중/실패까지 모두 끝내야 한다. 이는 클락 주파수가 낮은 CPU에 대해서는 적용 가능한 기술이지만 고속의 주파수를 가진 CPU의 경우 불가능하다. 이를 일반화한 경우가 필터 메커니즘으로 다수의 버퍼를 가지면서 접근 시간(access time)을 정상적인 한 사이클로 수행된다.

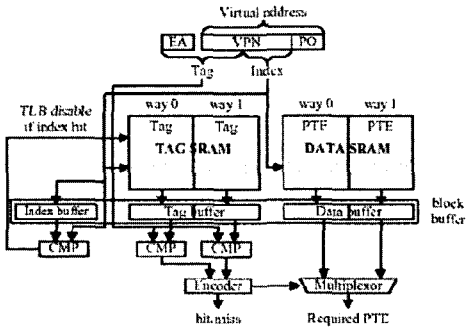


그림 1 블록 버퍼링 회로도

블록 버퍼링 메커니즘의 경우 캐쉬 메모리 보다 TLB에 적용할 경우 기대 효과가 더 크다. 즉 TLB의 경우 하나의 엔트리는 하나의 페이지(가령 4KB)를 나타냄으로써 일반적으로 하나의 페이지는 수십 또는 수백 개의 데이터 또는 명령어로 구성된다. 그럼으로 하나의 명령어가 참조되어질 경우 순차적으로 명령어가 인출될 확률이 높기 때문에 계속적으로 동일한 페이지를 접근할 확률이 높다. 그림 1은 블록 버퍼링을 이용한 2-way set associative TLB 구조이다.

3.2 필터 메커니즘(filter cache or micro TLB structure)

필터 메커니즘은 계층적인 메모리 시스템 구조로써 일반적인 레벨 1 캐쉬(L1 cache) 또는 주(main) TLB 위에 작은 크기의 필터 버퍼를 위치시킴으로써 시간적 지역성을 통한 소비 전력을 줄이는 방법이다[13]. 캐쉬 메모리 및 TLB 모두 이용 가능한 기술이지만 캐쉬의 경우 명령어 캐쉬에만 주로 이용 가능하다. 즉 명령어 캐쉬의 경우 적은 엔트리 개수를 가진 필터 버퍼의 적중률이 대단히 높기 때문에 성능과 소비 전력면에서 매우 우수한 성능을 보이고 있으나 데이터 캐쉬의 경우 필터 버퍼의 적중률 감소로 성능이 급격히 떨어지는 단점을 가진다. 그러나 TLB의 경우 명령어 및 데이터 모두 접근 실패율이 대단히 낮기 때문에 적은 엔트리를 가진 필터 버퍼를 이용함에도 불구하고 적중률은 대단히 높다.

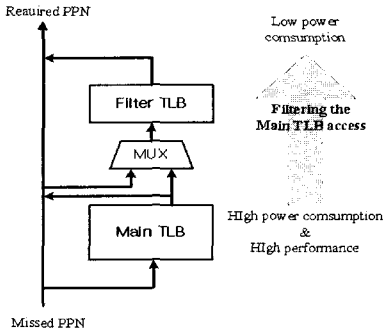


그림 2 필터(filter) TLB의 구조

대표적인 내장형 프로세서인 SH-4와 PA-RISC 2.0는 4개의 마이크로 명령어 TLB를 단일 TLB (Unified-TLB) 상위 계층에 위치시켜 명령어에 대해 한 사이클 내에 참조가 일어나고, 참조 실패인 경우 다음 사이클 동안 만일 TLB를 참조하는 메커니즘을 사용하고 있다. 그림 2는 이러한 필터 TLB의 개념적 구조를 보여준다. 작은 엔트리를 갖는 필터 TLB는

최근에 참조되어진 가상 페이지 번호(virtual page number: VPN)를 지님으로서 빠른 접근시간과 저전력의 성능을 보장해주고, 필터 TLB의 접근 실패인 경우 하단의 주 TLB로의 접근이 이루어져 일반적인 TLB의 성능을 유지시켜 준다.

3.3 뱅크 메커니즘(bank structure)

뱅크 메커니즘을 이용하는 캐쉬 또는 TLB는 온칩 메모리 참조 시 소비되는 전력을 줄이기 위한 방법으로 전체 온칩 메모리를 뱅크 구조로 나누는 것이다 [14]. 전체 태그 또는 데이터 부분을 2-뱅크 또는 4-뱅크로 나눌 경우 비트 라인과 워드 라인의 감소에 의한 전력 소비 감소 효과를 얻을 수 있으며, 특히 CAM(content addressable memory)를 이용하는 TLB의 경우 동시에 참조되는 태그 부분의 엔트리 수가 줄어들기 때문에 소비 전력을 크게 줄일 수 있다. 그러나 이러한 뱅크 메커니즘은 하나의 뱅크에 편중될 확률이 높음으로 다른 뱅크의 활용도 감소로 성능을 저하시키는 단점이 있다.

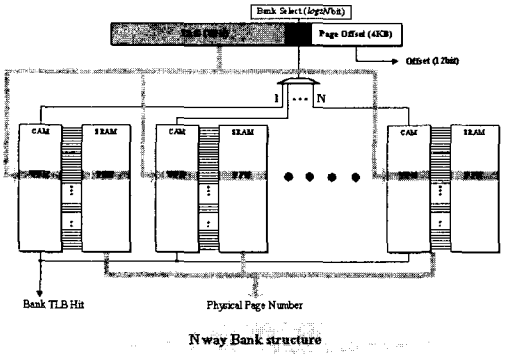


그림 3 N-way 뱅크 TLB 구조

그림 3은 N-way 뱅크 TLB 구조에 대한 회로도이며, 동작 메커니즘은 다음과 같다. 기존의 CAM 셀을 이용한 TLB를 N개의 작은 TLB로 분할하여 매번 참조 시 오직 하나의 작은 TLB만 활성화시켜 접근하는 방식으로, 요구된 가상주소의 페이지 오프셋 비트를 제외한 부분의 하위 $\log_2 N$비트가 뱅크 선택 비트가 되며, 이에 따라 오직 하나만의 뱅크만이 활성화된다. 그러므로 N개의 뱅크를 사용할 경우, 한번 접근 시 소비되는 전력량을 1/N으로 줄일 수 있는 장점이 있다.

4. 향상된 시스템 수준의 저전력 온칩 메모리 설계 기법

위에 제시된 시스템 레벨의 저전력 설계 기법은 제시한 바와 같이 여러 가지의 단점들을 가지고 있다. 이러한 단점들을 극복하고 보다 향상된 저전력 온칩 메모리 시스템 구조는 다음과 같다.

4.1 선택적 블록 버퍼링 기법(selective block buffering)

앞서 언급한 것처럼 블록 버퍼링은 고속의 클락 주파수를 가진 마이크로 프로세서의 경우 사용에 대한 제약을 받는다. 이를 해결하기 위한 효과적인 방법은 지능적/선택적으로 블록 버퍼 또는 주 온칩 메모리를 참조하도록 설계하는 것이다. 또한 단일 버퍼를 이용할 경우의 성능 저하를 막기 위하여 다중 뱅크 구조를 이용한다. 그림 4는 이러한 선택적 블록 버퍼링에 대한 TLB 구조이다.

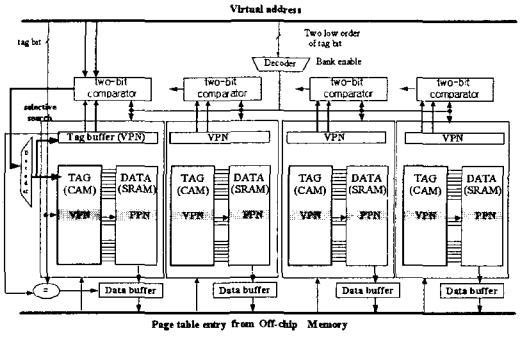


그림 4 선택적 블록 버퍼링 TLB 구조

선택적 블록 버퍼링 TLB 구성과 동작 메커니즘은 다음과 같다. 하나의 완전 연관 TLB의 태그는 CAM으로 구성되어지기 때문에 엔트리 수에 선형적으로 소비 전력이 증가한다. 이를 줄이기 위하여 기본 구조를 4개의 다중 뱅크로 구성하여 TLB 참조 시 소모되는 소비 전력을 70% 정도 줄일 수 있다. 또한 블록 버퍼링 기법을 이용하기 위하여 각각의 뱅크 위에 하나의 태그 버퍼를 위치 시켜 해당 뱅크에 대해 가장 최근에 참조되어진 주소의 가상 페이지 번호를 저장한다. 또한 선택적 블록 버퍼링 기법을 이용하기 위하여 2-bit 비교기를 각각의 뱅크 위에 위치시킨다.

예로 하나의 가상 주소가 생성되면 VPN의 최하위 2 비트를 이용하여 4개의 뱅크 중 하나를 선택한다. 이와 동시에 4개의 비교기는 블록 버퍼의 태그 버퍼에 저장된 VPN과 생성된 가상 주소의 VPN의 두 비트를 비교 수행한다. 물론 뱅크가 선택되어진 후 하나의 비교기만을 비교하여도 무방하지만 빠른 접근 시간을 보장하기 위하여 뱅크 선택과 동시에 4개의 비교기가 동시에 수행된다.

만약 해당하는 뱅크의 비교기에서 적중이 발생하면 생성된 주소와 해당 뱅크의 가장 최근에 참조되어진 가상 주소가 동일할 확률이 대단히 높음을 의미한다. 그러나 만약 2-비트 비교기에서 적중 실패이면 해당 뱅크에 대해 동일한 가상 주소가 연속적으로 접근하지 않았음을 의미한다. 이 경우 블록 버퍼의 태그 버퍼에 대한 접근 없이 한 사이클 동안 기존의 뱅크 TLB를 검색하게 된다. 만약 2-비트 비교기에서 적중이 발생하면 한 사이클 동안 블록 버퍼의 태그 버퍼를 참조하게 되고 태그 버퍼에서 적중이 발생하면 출력 드라이브인 데이터 버퍼에서 요청한 물리적 페이지 번호(physical page number: PPN)를 캐쉬에 보내게 된다. 그러나 2-비트 비교기에서 적중이 발생하였으나 블록 버퍼의 태그 버퍼에서 적중 실패가 일어나면 추가적인 한 사이클을 낭비하게 되고 다음 사이클 동안 기존의 뱅크 TLB를 검색하게 된다. 이와 동시에 블록 버퍼의 태그 버퍼는 생성된 주소의 VPN를 저장하여 연속적인 참조에 대비한다. 물론 2-비트 이상 비교를 수행할 경우 보다 효과적인 예측이 가능하지만 수행 시간이 길어지는 단점을 가지게 된다.

시뮬레이션의 결과에 따르면 Spec95 벤치마크의 경우 VPN의 하위 4번째 비트와 5번째 비트를 비교할 경우 그 예측 적중률은 96% 이상으로 두 사이클 오버해드는 단지 4%에 불과하다. 이는 동일한 구조를 가진 기존의 블록 버퍼링을 이용할 경우 22%에 비해 매우 낮은 수치임을 알 수 있다. 또한 블록 버퍼의 적중률은 80%, 뱅크 TLB의 적중률은 20%로 TLB에서 소비되는 전체 전력을 효과적으로 줄일 수 있음과 동시에 적은 성능 감소 효과를 얻을 수 있다.

4.2 뱅크 배치율 증대 기법

뱅크 메커니즘을 이용하는 메모리 구조의 경우에는 기존 구조의 접근 시간을 유지하면서, 전력을 줄이는데 효과적인 구조라고 할 수 있다. 그러나 뱅크 구조 메모리 시스템의 각 뱅크의 활용도는 CPU에서 발생되는 주소의 선택 비트(select bits)에 의해서 정해진다. 즉 뱅크의 사용이 특정한 하나의 뱅크로 편중되는 경우에는 각각의 뱅크로 균일하게 분포시키지 못하므로, 하나로 구성된 기존 구조에 비해 성능의 저하를 야기 시킨다. 예로, 2개의 뱅크 구조를 사용했을 경우, 뱅크0과 뱅크1의 접근율이 30:70 혹은, 20:80으로 나눠진 경우를 의미한다. 이러한 특정 뱅크로의 편중적인 접근의 문제는 온칩 메모리 시스템의 경우 뱅크 충돌 미스(conflict miss) 등의 문제를 발생시키며, 전체 엔트리를 비능률적으로 활용하는 단점을 보인다.

기존의 연구들은 이러한 문제를 해결하기 위해서, 뱅크를 선택해주는 부가적인 로직을 사용하여 해결하였다. 그러나 이러한 부가적인 함수 기능을 하는 로직의 경우는 추가적인 복잡도와 접근 시간의 지연을 초래함으로 이보다 단순한 하나의 XOR 게이트만을 사용하여 이 문제를 해결할 수 있는 방법이 제시되었다. 즉, VPN의 최하위 비트만을 이용하는 대신에 VPN의 특정한 위치의 비트들을 XOR을 시켜서 선택 비트를 보다 균일한 분포를 이루도록 만드는 것이다. 만약 두 개의 뱅크로 되어 있는 구조에서는, 선택 비트는 VPN의 최하위 한 비트가 된다. 이 선택 비트를 단순히 최하위 비트가 아닌 최하위 비트와 바로 인접해 있는 태그 비트 또는 다른 위치의 한 비트와 XOR 시킨다면, 보다 더 균일한 뱅크의 접근을 만들 수 있다. 이 기법은 XOR되는 비트를 늘일수록 더 균일한 분포를 얻을 수 있으나, 접근 시간에 대한 최소한의 부가적인 지연을 위해 선택 비트의 수보다 대략 2배수 정도일 때 최적화되어진다. 이러한 단순한 기법을 사용하여 저전력에 효과적인 뱅크 기법의 단점을 보완시키고 성능의 저하를 막을 수 있다.

4.3 필터 메커니즘과 뱅크 구조의 효과적인 조합

저전력 메모리 구조의 설계에 있어서, 위에서 기술되어진 필터 메커니즘과 뱅크 메커니즘 등의 적용은 설계적 접근 기법으로 매우 단순하면서도, 소비 전력 적인 측면에서 효과적인 기법들이라 할 수 있다. 그럼으로 두 기법의 단점을 보완하고 성능을 늘일 수 있는 새로운 합성구조에 대한 연구는 다음과

같다. 즉, 필터 메커니즘과 뱅크 구조의 효과적인 합성을 통한 통합 구조이다.

필터 기법의 경우에는 순수하게 4개 엔트리를 갖는 필터 TLB의 경우, 그 접근 성공률은 대략 80%를 보인다. 그러나 접근 실패가 일어난 나머지 20%의 참조는 주 TLB를 접근하기 위해 부가적인 수행 시간을 필요로 하는 오버헤드가 발생한다. 그럼으로 필터 버퍼의 접근 성공률을 높이기 위하여 하나의 태그 엔트리 당 두 개의 데이터 엔트리, 즉 서브 뱅크로 구성하는 방법이다. 이 방법은 동일한 CAM 전력 소모를 유지하면서 한 엔트리 당 더 많은 데이터를 지님으로 접근 성공률을 높인다. 또한, 전체 구조 역시 뱅크 기법을 사용하여 전체 소비 전력을 줄이고자 하였다.

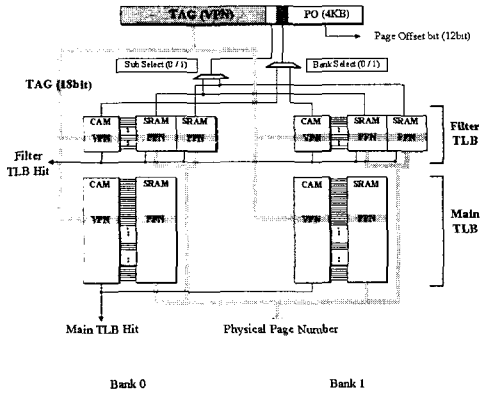


그림 5 필터 메커니즘과 뱅크 메커니즘의 합성 기법

그림 5는 이 합성 기법을 TLB구조에 적용시킨 예로서, 2개의 서브뱅크를 지닌 필터 TLB와 주 TLB를 나타낸다. 필터 TLB와 메인 TLB는 각각 뱅크 구조로 설계되어 있다.

위에서 살펴본 예의 동작을 살펴보면 다음과 같다. CPU가 특정 가상 주소를 발생시키면, 먼저 필터 TLB의 참조가 이루어진다. 뱅크 선택 비트는 가장 간단한 기법인 VPN 최하위 비트가 선택 비트가 되어 뱅크0 모듈 혹은 뱅크1 모듈 중 하나의 뱅크 모듈이 활성화된다. 동시에 선택된 뱅크 모듈의 서브 뱅크 선택 비트가 필터 TLB의 엔트리 중 두 개의 PPN 중에 하나를 선택한다. 접근이 성공일 경우, 해당되는 PPN이 캐쉬로 보내지게 된다. 만약 필터 TLB에서 접근이 실패인 경우, 다음 사이클에 주 TLB로의 접근이 이루어진다. 주 TLB의 접근 성공일 경우, 참조된 엔트리는 필터 TLB로 복사되고, 주 TLB의 해당 엔트리는 무효화(invalid) 된다. 일반적인 계층구조 메모리 구조에서는 하위 계층의 메모리의 내용은 반드시 상위계층의 내용을 포함하여 메모리 계층 구조간의 일치성(Consistency)을 보장해주어야 한다. 그러나 제시된 구조의 경우, 필터 TLB와 주 TLB간의 일치성을 제거 시켜 특정 엔트리의 중복을 피함으로서, 유효 엔트리 수를 증가시킬 수 있다. 만약 필터 TLB와 주 TLB내에서 모두 접근 실패되었을 경우에는 하위 메모리로 접근하여 참조를 반복하거나, 페이지 테이블 등을 통해 요구된 페이지를 다시 업로드 한다. 이 경우, 요구된 페이지는 주 TLB를 거치지 않고 직접 필터로 올라오며 필터에서 대치(replacement) 되어지는 엔트리는 주 TLB로 복사되는 기법을 사용하였다

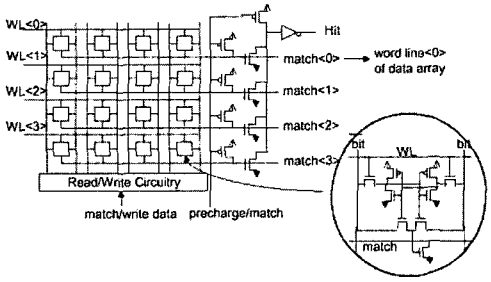


그림 6 일반적인 CAM 셀 구조

5. 회로 레벨의 저전력 온칩 메모리 설계 기법

일반적으로 회로 레벨 관점에서 보면 SRAM에 비해 CAM의 소비 전력이 대단히 높다. CAM은 그림 6에서 보듯이 6-트랜지스터 SRAM과 달리 외부 입력 데이터와 셀에 저장되어 있는 데이터간의 비교를 수행하기 위하여 XOR를 구성하는 두 개의 트랜지스터와 매치 라인을 구동하기 위한 별도의 트랜지스터가 추가된다. 원하는 PTE(page table entry)의 태그 주소가 CAM 태그 메모리로 입력되면 태그 메모리의 모든 태그 주소와 입력 태그간에 동시에 비교 검사가 수행되고 만일 하나의 태그주소 일치가 발생할 경우 해당하는 매치 라인(match line)은 충전(charge) 상태로 있고, 일치하지 않은 나머지 태그 주소에 해당하는 매치 라인은 방전(discharge)된다. 선택되어진 매치 라인은 데이터가 저장되어 있는

SRAM의 특정한 워드라인을 구동하게 되고 해당하는 PTE 정보가 데이터 출력 버퍼로 읽혀지게 된다. 완전 연관 방식의 TLB에는 태그 비교를 위한 외부 비교기나 멀티플렉서가 필요 없으나 CAM내에서 이루어지는 태그의 비교 검사와 SRAM에서 데이터가 읽혀지는 동작이 순차적으로 발생할 수밖에 없기 때문에 접근 시간이 직접 사상이나 집합 연관 방식보다 길어진다.

캐쉬 블록과 마찬가지로 TLB 블록의 전력 소모는 대부분 메모리 셀의 워드 라인과 비트 라인을 구동하기 위해서 충전(charge) 혹은 방전(discharge)되는 정전 용량(Capacitance)에 의한 것이다. CAM에서는 매 접근마다 모든 비트 라인과 매치 라인이 미리 충전(precharge)되어 있어야 하며 비교 후 일치하지 않는 워드들의 모든 매치 라인은 방전되어야 한다. 이 같은 회로의 동작 특성은 CAM이 SRAM에 비해 더 많은 전력을 소모하게 하는 주요 원인이다.

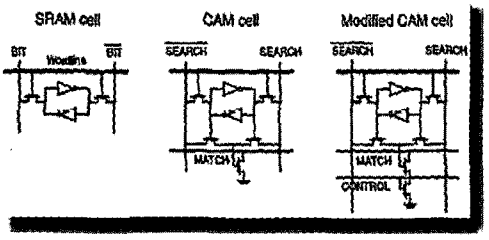


그림 7 수정된 CAM 셀 구조

이러한 CAM의 높은 소비 전력을 줄이고자 그림 7과 같이 수정된 CAM 셀이 제시되었다. 즉, 매치 라인 밑에 추가적인 트랜지스터를 삽입함으로써 일치하지 않은 나머지 태그 주소에 해당하는 매치 라인의 방전을 없애게 하였다. 이러한 효과는 CAM 셀의 소비 전력을 약 40% 정도 줄일 수 있다.

이외에도 SRAM에서 소비되는 전력을 줄이고자 많은 연구들이 활발히 진행중이며 효과적인 구조가 많이 제안되고 있다.

6. 결 론

저전력 온칩 메모리에 대한 연구는 고성능 메모리 시스템 설계 기법과 더불어 오늘날 중요한 요인중의 하나로써 부각되고 있다. 저전력/고성능 프로세서를 설계하고자 할 경우 고려해야할 중요 요인 중의 하나가 바로 캐쉬 메모리와 TLB이다. 그러나 TLB가 캐쉬와 더불어 전력 소모가 많은 블록임에도 불구하고 지금까지 저전력 메모리 시스템에 대한 연구는 주로 캐쉬 메모리에 초점을 맞추어 진행되어 왔다. 이에 제시된 기법들은 캐쉬 메모리뿐만 아니라 특히 TLB에 효과적인 저전력 기법을 중심으로 소개하였다. 가장 대표적인 시스템 레벨의 저전력 기법은 블록 버퍼링 메커니즘, 뱅크 메커니즘, 그리고 필터 메커니즘을 효과적으로 이용하는 것이다. 그러나 기존의 이러한 메커니즘들은 성능 저하라는 단점을 내포하고 있지만 소비 전력 적인 측면에서는 대단히 우수한 기법들이다. 이에 이러한 기법들의 단점을 극복하고 보다 효과적인 방법으로 접근한다면 저전력 메모리 시스템 설계에 좋은 지침이 될 수 있을 것이다.

참고문헌

[1] S. Santhanm, "StrongARM SA110, a 160mhz 32b 0.5w CMOS ARM processor," in Proc. Hot Chips 8, Aug. 1996.

[2] F. J. Sanchez, A. Gonzalez and M. Valero, "Static Locality Analysis for Cache Management," Technical report UPC-DAC-1997-28.

[3] A. J. Smith, "Cache Memories," ACM Computing Surveys, vol. 14, no. 3, pp. 473-530, Sep. 1982.

[4] T. M. Austin and G. S. Sohi, "High-bandwidth Address Translation for Multiple-issue Processors," in Proc. the 32rd Intl. Symp. on Computer Architecture, pp. 158-167, May 1996.

[5] M. Talluri, S. Kong, M. D. Hill and D. A. Patterson, "Tradeoffs in Supporting Two Page Sizes," in Proc. the 19th Intl. Symp. on Computer Architecture, pp. 415-424, May 1992.

[6] Y. A. Khalidi, "Virtual Memory Support for Multiple Page Sizes," in Proc. the 4th Workshop on Workstation Operating Systems, Oct. 1993.

[7] T. Juan, T. Lang and J. Navarro, "Reducing TLB Power Requirements," in Proc. Intl. Symp. on Low Power Electronics and Design, 1997.

[8] 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.

[9] T. Pering, T. Burd and R. Brodersen. "Dynamic Voltage Scaling and the Design of a Low-Power Microprocessor System," in Proc. Power Driven Microarchitecture Workshop, attached to ISCA98, June 1998.

[10] M. B. Kamble and K. Ghose, "Energy-Efficiency of VLSI Cache: A Comparative Study," in Proc. the 10th. Intl. Conf. on VLSI Design, pp. 261-267, Jan. 1997.

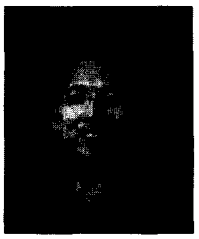
[11] M. B. Kamble and K. Ghose, "Analytical Energy Dissipation Models for Low Power Caches," in Proc. Intl. Symp. on Low-Power Electronics and Design, Aug. 1997.

[12] K. Ghose, and M. B. Kamble, "Reducing Power in Superscalar Processor Caches using Subbanking, Multiple Line Buffers and Bit-line Segmentation," in Proc. Intl. Symp. on Low-Power Electronics and Design, pp. 70-75, Aug. 1999.

[13] Kin, et. al., "Filtering Memory References to Increase Energy Efficiency," IEEE Transactions on Computers, vol. 49, no. 1, Jan. 2000.

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

이 정 훈



1999 성균관대학교 제어계측공학과(학사)
2001 연세대학교 컴퓨터과학과(석사).
2001~현재 연세대학교 컴퓨터과학과 (박사과정)
관심분야:지능형 메모리 시스템, 저전력-고성능 시스템, 고성능 컴퓨터 구조임
E-mail:ljh@yonsei.ac.kr

최 진 혁

2001 한양대학교 전자·컴퓨터공학부 (학사)
2001~현재 연세대학교 컴퓨터과학과 (석사과정)
관심분야:저전력 메모리 시스템, 고성능 컴퓨터 구조임
E-mail:jhchoi@cs.yonsei.ac.kr

김 신 덕

1982 연세대학교 공과대학 전자공학과 (학사)
1987 University of Oklahoma 전기공학 (석사)
1991 Purdue University 전기공학(박사)
1993~95 광운대학교 컴퓨터공학과 조교수
1995~현재 연세대학교 공과대학 컴퓨터과학과 교수
관심분야:고성능 컴퓨터 시스템, 웹 컴퓨팅임
E-mail:sdkim@cs.yonsei.ac.kr