

4Ghz 고성능 CPU 위한 캐시 메모리 시스템

정 보 성*, 이 정 훈**

Cache memory system for high performance CPU with 4GHz

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

요 약

본 논문에서는 4Ghz의 빠른 클럭 속도의 CPU에 적합한 고성능 L1 캐시 메모리 구조를 제안한다. 제안된 캐시 메모리는 빠른 접근 시간을 위한 직접사상 캐시와 시간적 지역성을 고려한 2-way 연관사상 버퍼 그리고 버퍼 선택 테이블로 구성된다. 빠른 접근 시간을 보장하는 직접사상 캐시는 가장 최근 접근한 데이터를 저장하게 된다. 만약에 직접사상 캐쉬로부터 추출되는 데이터가 다시 참조되어질 높은 확률을 가지는 데이터이면 그 데이터들은 2-웨이 연관사상 버퍼로 선택적으로 저장되어 진다. 그리고 고성능과 저전력의 효과를 높이기 위하여 2-웨이 연관사상 버퍼중 하나의 웨이만 선택적으로 먼저 접근되어지며, 이러한 동작은 버퍼 선택 테이블에 의해 선택된다. 시뮬레이션 결과에 따르면, 에너지 소비와 평균 메모리 접근 시간을 고려한 에너지*지연시간에서 두배 이상의 크기를 가지는 직접사상 캐시, 4-웨이 연관사상 캐시 그리고 희생 캐시에 비해 각각 45%, 70% 그리고 75%의 성능향상을 이루었다.

▶ Keywords : 고성능 데이터 캐시 메모리, 동적 메모리 접근 시간, 저전력

Abstract

In this paper, we propose a high performance L1 cache structure on the high clock CPU of 4GHz. The proposed cache memory consists of three parts, i.e., a direct-mapped cache to support fast access time, a two-way set associative buffer to exploit temporal locality, and a buffer-select table. The most recently accessed data is stored in the direct-mapped cache. If a data has a high probability of a repeated reference, when the data is replaced from the direct-mapped cache, the data is selectively stored into the two-way set associative buffer. For the high performance and low

• 제1저자 : 정보성 • 교신저자 : 이정훈

• 투고일 : 2012. 12. 5, 심사일 : 2013. 1. 9, 게재확정일 : 2013. 1. 22.

* 국립 경상대학교 제어계측공학과(Dept. of control and Instrumentation Engineering, Gyeongsang National University)

** 국립 경상대학교 제어계측공학과(Dept. of control and Instrumentation Engineering, Gyeongsang National University)

power consumption, we propose an one way among two ways set associative buffer is selectively accessed based on the buffer-select table(BST). According to simulation results, Energy * Delay product can improve about 45%, 70% and 75% compared with a direct mapped cache, a four-way set associative cache, and a victim cache with two times more space respectively.

▶ Keywords : Data cache memory, Dynamic access times, low power consumption

I. 서 론

유아날날, 대용량 및 복잡한 데이터 처리를 위해 컴퓨팅 시스템은 더욱 빠른 처리 속도를 위해 빠른 클럭 속도의 프로세서를 요구하고 있다. 실제, Intel의 경우 i5는 기본 클럭 3.3Ghz에서 자동 오버클럭 기술을 이용하여 최대 3.7Ghz까지를 CPU가 보장하고 있다(1). 또한 메모리의 집적도 및 접근 속도 역시 급격하게 발전하고 있다(2). 비록, 메모리의 접근 속도가 향상되었다 하여도, 여전히 빠른 클럭 속도의 프로세서와 메모리의 속도의 차이는 시스템의 성능저하의 원인이 되고 있다.

빠른 클럭 속도는 높은 전력 소비와 발열에 의한 제한적인 클럭 속도를 제공하였고, 이를 해결하기 위해 공정의 기술 개발(22nm~)등 다양한 요소로 현재 클럭 속도는 급속도로 증가하며, Moor's low를 뒷받침하고 있다.

비록 빠른 데이터 처리를 위해 빠른 클럭 속도의 프로세서가 제공된다 하여도 여전히 칩 소비파워 및 발열현상은 심각한 문제이다. 더욱이, 빠른 클럭 속도의 프로세서는 칩 내부 캐시 메모리에 대한 추가적인 접근 시간을 요구하게 된다.

예로, StrongARM SA 110(3)의 경우, 전체 칩의 43%가 칩 내장 캐시 메모리의 소비전력을 소비하고 있으며, 또한 고성능 컴퓨팅 시스템에서는 높은 웨이 연관사상 캐시 메모리(1) 사용하고 있다. 따라서 앞으로 빠른 클럭 속도의 프로세서의 등장은 칩 내장 캐시 메모리의 더 높은 소비전력과 추가적인 메모리 접근 시간이 소비될 것이다.

컴퓨팅 시스템에서 캐시 메모리는 전체 시스템의 성능향상 및 저전력을 위한 중요한 메커니즘으로 자리 잡고 있다. 캐시 메모리의 성능향상을 위해 대표적인 방법으로 구조적인 모델링 방법으로 프로그램 수행 시 적합한 지역성(4) 사용과 충돌 접근 실패율을 줄이는 방법이다. 하지만, 빠른 클럭 속도의 프로세서에서는 프로그램 수행 시 적합한 지역성을 효과적으로 사용하기 위해 시간 지역성은 많은 엔트리(entry) 수를,

공간 지역성은 큰 블록 용량을 요구하며, 충돌 접근 실패에 효과적인 n-웨이 연관사상 캐시 혹은 회생 캐시 메모리 경우 구조인 접근에서 추가적인 접근 시간뿐 아니라 역시, 높은 소비파워의 원인이다.

현재 Intel은 트라이게이트 트랜지스터 기술을 바탕으로 22nm공정뿐 아니라 차후 14nm, 5nm의 공정을 계획하고 있다. 따라서 앞으로 고성능 데이터 처리를 위한 빠른 클럭의 프로세서가 등장 할 것이며, 내장형 캐시 메모리와 CPU의 접근 속도 역시 점점 그 차이가 커질 것이다.

표 1.은 현재 가장 기본적으로 사용되는 내장형 캐시 메모리의 1Cycle 접근 경계 속도를 나타낸 표이다(5). 현재 고성능 프로세서의 경우, 32KByte~64KByte의 4웨이 연관사상 캐시 메모리를 사용하고 있으며, 이는 표 1.과 같이 2~3 Cycle의 접근 속도를 가진다. 이에 빠른 클럭 속도를 가지는 고성능 프로세서의 등장은 더 높은 내장형 캐시 메모리의 접근 시간을 요구하게 될 것이다. 따라서 본 논문에서는 고성능 프로세서에 적합한 내장형 캐시 메모리의 구조를 제안하였다. 본 논문에서는 제안된 구조를 위해 4GHz의 클럭 속도를 가지는 프로세서로 가정하였으며, 이데 빠른 접근 속도와 저전력을 고려한 구조 표 1.를 참조로 선택하였다.

표 1. 다양한 캐시메모리의 1Cycle 접근 경계 속도(GHz)
Table 1. the various cache memory Boundary speed for 1Cycle access times(GHz)

항목	DM	2W	4W	FA	
8KByte	4.30	3.45	3.27	1KByte	2.59
16KByte	3.78	2.91	2.89		
32KByte	3.52	2.24	2.73	2Kbyte	2.26
64Kbyte	3.13	2.12	2.09		

II. 관련 연구

오늘날, 컴퓨팅 시스템에서 캐시 메모리는 성능향상 및 저

전력을 위해 가장 중요한 역할 뿐 아니라, 그 비중이 점점 커지고 있다. 이에 따라 저전력 및 고성능 내장형 캐시 메모리에 대한 다양한 연구가 이루어지고 있다.

Way-Cache(WC)[6]의 연구는 CAM(Contents Address Memory)를 이용하여 최근 접근된 연관사상 캐시 메모리들 중 하나를 선택하여 빠른 접근 시간과 낮은 메모리 소비전력을 향상시킨 연구이다. WC는 LSQ(load/store Queue)에 저장된 주소에 대해 웨이-연관사상 캐시 메모리의 웨이 위치, 주소 그리고 유효 비트로 구성되며, WC에서 접근 성공이면, 해당 n-웨이 연관사상 캐시 메모리 중 한 웨이를 선택하여 접근하게 된다.

WTSA cache[7]는 Way-Tracking Table(WTT)을 이용하여 연관사상 캐시 메모리를 선택적으로 접근하여 성능과 소비파워에 대한 향상을 이루었다. WTT는 Tag의 하위 비트를 이용하여 현재 하위 비트가 포함되는 웨이만 접근을 하게 된다.

하지만, WC의 경우, LSQ에 없는 데이터에 대해서는 웨이-연관사상 캐시에 대한 보장을 할 수 없고, LSQ의 내용보다 캐시 메모리의 데이터가 먼저 교체되면 캐시 메모리의 접근 실패가 발생하게 된다. WTSA의 경우, 기존 연관사상 캐시 메모리에 비해 접근 웨이를 줄이므로 소비 파워에 효과적인 면을 보이지만, 평균 메모리 접근 시간과 접근 실패율에서는 기존 연관사상 캐시 메모리에 대해 높은 성능향상을 개선할 수가 없다. 더욱이, 연관사상 캐시 메모리의 접근을 위해서는 WTT를 접근 후 캐시 메모리의 접근으로 실제 추가적인 메모리 접근 시간이 소비되어진다.

WSSA cache는[8] 연관사상 캐시의 특정 블록의 접근 빈도를 고려하여 웨이의 블록을 공유하므로 성능향상을 이루었으며, Set-Balancing Cache[9]는 빈번하게 접근 되는 블록의 접근 실패 시 충분히 사용되지 않는 블록에 데이터를 저장하게 된다. 하지만, WSSA cache와 Set-Balancing Cache는 기존 연관사상 캐시 메모리와 동일한 접근 시간을 가지며, 캐시 메모리 접근 실패시 공유 블록 혹은 다른 관련 블록의 접근을 위한 추가적인 접근 시간을 요구하게 된다.

SWA cache[10]는 기존 2-웨이 연관사상 캐시 메모리에서 가장 최근에 접근 한 블록을 순차적으로 접근 하여 소비파워 및 평균 메모리 접근 시간을 향상 시켰다. 비록 SWA가 소비파워 및 메모리 접근 시간을 향상 시켰지만, 여전히 접근 실패율은 기존 2-웨이 연관사상 캐시 메모리의 실패율을 보이며, 이는 실제 2-웨이 연관사상 캐시 메모리에 한정된 구조라고 할 수 있으며, 연관사상 캐시의 웨이가 증가 할수록 이 구조는 오히려 성능저하를 야기 시킬 수 있다.

본 논문을 위해 선행 연구로 기존 웨이 연관사상 캐시(2-웨이, 4웨이)에 대한 가장 최근에 참조된 블록의 재 참조 비율을 모니터링 하였다. 16KByte의 용량의 웨이 연관사상 캐시 메모리에서 약 75%이상이 최근 참조된 데이터를 재 참조하는 것을 확인하였다. 이를 바탕으로 본 논문에서는 빠른 CPU 클럭 속도를 가지는 고성능 내장형 캐시 메모리를 제안하였다.

III. 본 론

3.1 고성능 캐시 메모리 제안 동기 및 방법

본 논문의 주목적은 오늘날 급속하게 증가하는 CPU 클럭 속도에 적합한 효과적이고 빠른 내장형 캐시 메모리의 설계에 그 목적을 두고 있다.

오늘날, 비록 메모리의 접근 시간이 더욱 개선되었다하여도, 여전히 CPU와 메모리의 접근 시간은 전체 시스템의 성능향상에 중요한 요인으로 자리하고 있다. 현재 컴퓨팅 시스템에서 내장형 캐시 메모리는 시스템의 빠른 접근 시간과 저전력으로 전체 시스템의 성능향상의 가장 중요한 역할을 수행하고 있다.

캐시 메모리의 효과적인 성능향상을 이루는 방법으로 프로그램 수행 시 적합한 두 지역성(시간/공간 지역성)을 이용하는 것과 캐시 메모리의 충돌 접근 실패를 줄이는 방법이 가장 대표적인 방법이다.

프로그램 수행 시 적합한 두 지역성으로, 공간 지역성은 '참조 되어진 블록의 이웃 블록에 참조될 확률이 높다'는 것이며, 시간 지역성은 '한번 참조되어진 블록은 짧은 시간에 다시 참조 될 확률이 높다'는 것을 의미한다. 따라서 공간적 지역성은 블록 크기가 클수록, 시간 지역성은 블록 개수가 많을수록 효과적인 성능향상을 이룰 수 있다. 하지만, 블록의 크기 및 개수의 증가는 실제 접근 시간 및 소비파워의 증가를 가져온다.

캐시 메모리의 접근 실패를 줄이는 가장 좋은 방법으로 현재 일반적으로 상용화된 구조는 연관사상 구조이다. 연관사상 구조는 높은 연관사상을 가질수록 효과적인 성능향상을 보인다. 현재 일반적으로 내장형 캐시 메모리는 고성능을 위해 이러한 연관사상 구조를 채택하고 있다.

하지만, 현재 범용 컴퓨터 시스템의 CPU는 점점 고속의 클럭 속도(3.3Ghz~)를 이루고 있으며, 임베디드 시스템 역시 이전에 비해 빠른 클럭 속도(~ 2Ghz)로 빠른 데이터 처

리를 위한 고성능을 추구하고 있다. 공정 기술의 발전으로 범용 컴퓨팅 시스템 뿐 아니라 임베디드 시스템의 클럭 속도는 고성능을 위해 더욱 빠른 클럭 속도를 가질 것이다. 기존의 내장형 캐시 메모리가 메인 메모리에 비해 빠른 접근 시간(1cycle 접근 시간)을 보장하지만, 빠른 클럭 속도를 가지는 CPU에 대해 기존 상용화된 캐시 메모리는 추가적인 접근 시간을 가진다.

본 논문에서는 빠른 CPU 클럭 속도에 대한 성능향상에 효과적인 내장형 캐시 메모리 구조를 제안하였다. 제안된 캐시 메모리는 빠른 클럭 속도를 가지는 CPU(4Ghz)에 대해 평균 메모리 접근 시간에 효과적인 구조를 선택하기 위해 표 1의 캐시 메모리 접근 시간 경계를 기준을 선택하였다.

직접사상 캐시 메모리는 빠른 접근 시간을 보장하지만, 다른 캐시 메모리 구조에 비해 높은 충돌 접근 실패율을 가지는 단점을 보인다. 따라서 직접사상 캐시 메모리의 효과적인 성능향상을 위해서 Victim 캐시 메모리와 같이 희생 버퍼를 사용하는 것이 효과적인 방법이다. 접근 실패율에서 가장 간단 하면서 효과적인 성능향상을 보이는 구조는 연관사상 구조이다. 하지만 연관사상 구조는 높은 연관사상을 가질 때, 소비 파워 뿐 아니라 접근 시간까지 높아지는 단점을 가진다.

제안된 캐시 메모리는 빠른 접근 시간이 보장되는 직접사상 구조의 주 캐시 메모리와 접근 실패율에 효과적인 연관사상 구조의 희생버퍼로 구성되어진다. 주 캐시 메모리는 현재 참조가 일어난 데이터를 가지므로 빠른 접근 시간을 보장하며, 연관사상 버퍼는 참조 가능성이 높은 데이터를 저장하므로 성능향상을 이루었다.

연관사상 버퍼는 참조 가능성이 높은 웨이 블록을 순차적으로 접근이 이루어지며, 제안된 버퍼 선택 테이블(Buffer_Select_Table, BST)의해 접근 하며, 주 캐시 메모리의 접근동안 한 웨이 접근을 위한 연관사상 버퍼의 웨이를 선택하게 된다. 따라서 제안된 캐시 메모리는 참조 가능성이 높은 블록들을 순차적으로 접근하여 다양한 접근 시간을 가질 수 있다.

3.2 제안된 캐시 메모리 구조 및 동작

본 논문에서 제안된 캐시 메모리는 빠른 클럭 속도를 가지는 CPU에 대해 성능향상을 위해 세부분으로 구성되어진다. i)캐시 메모리의 빠른 접근을 위해 주 캐시 메모리는 직접사상 캐시 메모리와 ii)직접사상 캐시의 접근 실패율을 줄이기 위한 n-웨이 연관사상 버퍼 그리고 iii) n-웨이 연관사상 버퍼의 접근 순위와 캐시와 버퍼의 접근 실패 시 대체 블록 선정을 위한 버퍼 선택 테이블(BST)로 나누어진다.

제안된 캐시 메모리의 구조는 그림 1과 같다. 주 캐시 메모리는 CPU의 클럭 속도(4Ghz로 가정)에 1Cycle 접근을 위한 캐시 메모리 크기를 가진다. 또한 주 캐시 메모리인 직접사상 캐시 메모리는 n-웨이 연관사상 버퍼로부터 시간 지역성을 고려한 다시 참조 되어진 데이터를 판단하기 위해 추가적인 1비트(temporal bit-TB)를 가지고 있다. TB는 n-웨이 연관사상 버퍼에서 접근 성공 후 주 캐시 메모리로 이동할 때 주 캐시 메모리에 '1'로 갱신되어진다.

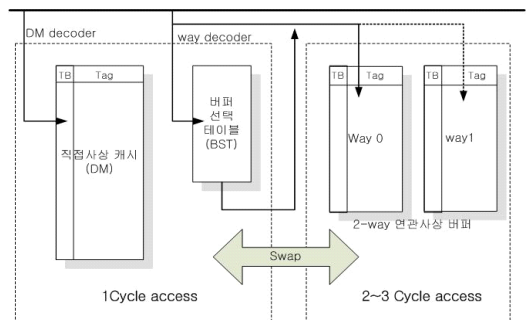


그림 1. 시스템 구조
Fig. 1. System Architecture

제안된 n-웨이 연관사상 버퍼는 주 캐시 메모리와 동일하게 시간 지역성을 위한 추가적인 1비트(TB)를 가진다. 그리고 직접사상 캐시 메모리에서 접근 실패가 발생하면, 현재 참조 가능성이 높은 웨이부터 순차적으로 접근하게 된다. 본 논문에서는 연관사상 버퍼의 순차적인 접근으로 효과적인 활용을 위해 2-웨이 연관사상 버퍼를 사용하였다. 2-웨이 연관사상 버퍼중 하나의 웨이 버퍼에서 접근 성공이 발생하면, 그 데이터는 주 캐시 메모리인 직접사상 캐시 메모리와 스왑(Swap)동작이 발생하게 된다. 데이터 스왑시 주 캐시 메모리의 TB는 2-웨이 연관사상 버퍼에서 선택된 웨이 블록의 TB로 갱신이 된다.

2-웨이 연관사상 버퍼의 순차적인 접근은 BST의 값으로 결정되며, BST는 연관사상 버퍼의 LRU 비트와 연관사상 버퍼의 각각의 웨이에 저장된 태그 최하위 1비트가 저장된 인덱스 테이블(Index table, IT)로 구성된다.

인덱스 테이블은 직접사상 캐시 메모리의 접근 인덱스와 현재 접근할 연관사상 버퍼의 인덱스가 동일한 인덱스인지 판별하게 되며, 연관사상 버퍼 중 직접사상 캐시 메모리와 동일한 웨이 블록만 접근하게 된다. 만약, 모든 웨이가 동일한 직접사상 캐시 메모리의 인덱스 값을 가진다면, 연관사상 버퍼는 LRU 동작에 의해 연관사상 버퍼중 하나의 웨이 블록에 접근하게 된다. 반면 모든 연관사상 버퍼가 직접사상 캐시 메모리

모리의 인덱스와 다르다면, 추가적인 연관사상 버퍼에 접근 없이 접근 실패가 이루어진다.

그림 2는 2-웨이 연관사상 버퍼의 접근을 나타낸 그림이다. 그림 2와같이 만약 CPU가 요청한 주소가 1001001이고, 직접사상 캐시와 2-웨이 연관사상 버퍼가 동일한 캐시 메모리 크기를 가질 때, 직접사상 캐시 메모리의 태그가 100, 인덱스가 1001 경우, 연관사상 버퍼의 태그는 1001 그리고 001의 인덱스를 가진다. 연관사상 버퍼의 태그 최하의 1비트("1")를 이용하여 IT에 접근하고 동일한 값이 존재하는 웨이의 블록에 접근한다(a). 만약 각 웨이의 값이 동일하다고 태그 최하위 비트와 같다면 연관사상 버퍼는 LRU 비트에 의해 웨이가 선택되며(b), IT가 태그의 최하위 비트와 다르다면 바로 접근 실패로 하위 계층의 메모리에 접근하게 된다(c).

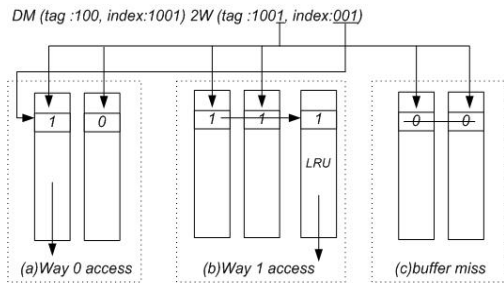


그림 2. 연관 버퍼 접근을 위한BST 동작 알고리즘.
Fig 2. BST operation algorithm for Set-associative buffer access.

이러한 방법은 간단하지만, 실제 연관사상 버퍼의 접근을 효과적으로 줄일 수 있다. 본 논문에서 시뮬레이션 결과 제안된 2-웨이 연관사상 버퍼의 접근 방법으로 2-웨이 연관사상 버퍼 접근 성공시 평균 88%가 하나의 버퍼 웨이만 접근한 것으로 나타났다. 또한 전체 접근 실패시 평균 23%가 직접사상 캐시 메모리에만 접근하였고, 58%가 2-웨이 연관사상 버퍼의 한 웨이만을 접근한 후 접근 실패가 발생한 것으로 나타난다. 이는 실제 연관사상 버퍼의 가장 큰 문제점인 접근 시간과 소비 파워를 많이 줄일 수 있는 방법이다.

본 논문에서 제안된 방법은 다음과 같이 요약 할 수 있다.

- I) CPU로부터 데이터가 요청되면, 직접사상 캐시 메모리와 BST에 동시에 접근이 일어난다. 직접사상 캐시 메모리에 접근 성공이면, 직접사상 캐시로부터 요청된 데이터를 CPU에 데이터를 전달하게 된다. 만약 직접사상 캐시가 접근 실패면, BST는 IT와 LRU비트로 2-웨이 연관사상 버퍼중 단일 웨이 혹은 순차적 접근 그리고 접근 실패를 선택하게 된다.
- II) BST에 의해 선택된 접근 방법에 의해 2-웨이 연관사

상 버퍼에 접근이 발생하고, 만약 접근 성공이면, 직접사상 캐시 메모리와 접근 성공이 이루어진 웨이의 데이터는 스왑이 이루어진다. LRU, IT, TB가 갱신되어진다. 선택되어진 웨이 블록의 TB는 직접사상 캐시 메모리의 TB로 갱신되며, 직접사상 캐시 메모리의 TB는 '1'로 갱신되어진다. 또한 IT 역시 웨이 블록에 들어오는 태그의 최하위 1비트로 갱신되어진다.

III) 만약, 직접사상 캐시 메모리와 2-웨이 연관사상 버퍼의 접근이 모두 실패일 경우, CPU가 요청한 데이터는 하위 계층 메모리에서 데이터를 가져오게 되며, 이때 주 캐시 메모리의 블록의 데이터가 유효하다면, 2-웨이 연관사상 버퍼에 저장하게 된다. 2-웨이 연관사상 버퍼의 희생 블록의 선택은 각 웨이 블록의 TB에 의해 결정하게 된다. 만약 두 웨이 블록의 TB가 동일하다면, LRU에 의해 희생 블록을 선택하며, 그렇지 않다면 TB가 '0'인 웨이의 블록을 희생 블록이 선택된다.

3.3 성능평가

본 논문에서 제안된 캐시 메모리의 구조의 성능평가로 접근 실패율(miss ratio), 평균 메모리 접근 시간(Average Memory Access Time) 그리고 소비파워와 평균 메모리 접근 시간을 고려한 에너지*지연시간(energy delay product)을 지표로 하였다. 표 2는 제안된 캐시 메모리의 성능 평가를 위한 시스템 환경 변수를 나타낸 표이다.

표 2. 시스템 환경
Table 2. System Environment

항목	값
CPU 처리 속도	4Ghz
직접사상 캐시 접근	1CPUcycle
2웨이 연관사상 버퍼 접근	2~3/CPUcycle
버스 폭	32bit
메인 메모리 접근 시간	168cycle/CPUcycle

제안된 캐시 메모리의 평가를 위해 본 논문에서는 CPU SPEC 2006[11][12]를 pin-tools[13,14] 시뮬레이터를 이용하여 참조 주소를 모니터링 하였다.

본 논문에서 메인 메모리는 다른 요소들은 배제한 CPU 사이클에 대한 전체 메모리의 최소 동작 사이클을 사용하였다. 메모리 계층은 제안된 구조의 성능 비교를 위해 CPU, L1 캐시 메모리 그리고 메인 메모리로 가정하였다. 제안된 캐시 메모리의 성능 평가를 위해 본 논문에서는 비교 캐시 메모리로 오늘날 가장 상용화된 구조의 캐시 메모리에 대해 성능 평가를 하였다. 오늘날 가장 많이 상용화된 구조로 직접사상 캐시(DM), 4-웨이 연관사상 캐시(4W) 그리고 Victim

캐시(VT) 메모리가 있다. 또한 제안된 구조와 비슷한 순차적인 접근이 이루어지는 SAC 캐시 메모리역시 대조 캐시 메모리로 선택하였다. 캐시 메모리의 용량은 DM_64KB, SCA_16KB, 4W_32KB, VT_32KB 그리고 제안된 캐시 메모리(Pro) 16KB를 사용하였다.

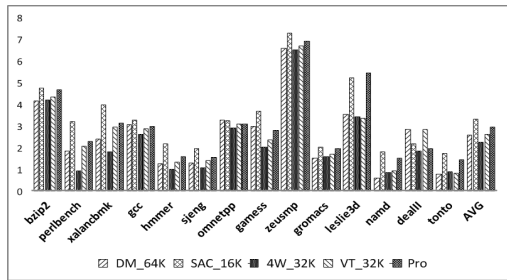


그림 3. 메모리 접근 실패율(%)
Fig 3. Memory Miss ratio(%)

그림 3은 캐시 메모리 접근 실패율을 나타낸 그림이다. 비록 제안된 캐시 메모리의 구조가 SCA 구조를 제외한 나머지 구조에서는 높은 접근 실패율을 보이고 있다. 이는 우선적으로 캐시 메모리의 구조적인 문제를 제외하여도 2배에서 4배 가량의 용량적인 차이가 있기 때문이다. 더욱이 구조적인 측면에서 우수한 4W와 VT 캐시구조는 더욱 좋은 성능을 보인다. 시뮬레이션 결과 4W와 VT 캐시보다 제안된 구조가 24%와 12%의 높은 접근 실패율을 보이지만, 평균 메모리 접근시간에서 제안된 캐시 메모리는 4W비해 단 6%의 성능저하를 가진다.

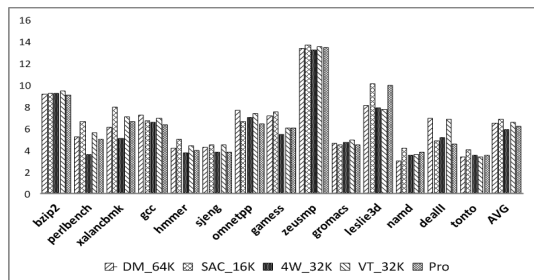


그림 4. 평균 메모리 접근 시간(cycle).
Fig 4. Average Memory Access Time(cycle).

그림 4는 평균 메모리 접근 시간을 나타낸 그림이다. 그림에서 보듯이 제안된 구조는 4W를 제외한 나머지 구조에 비해 효과적인 성능을 보이고 있다. 비록 DM_64K와 VT_32K가 접근 실패율에서 효과적인 성능을 보이지만 접근 실패율에서

오려려 약 5%의 낮은 평균 메모리 접근 시간을 가진다. 이는 제안된 캐시 메모리 구조는 작은 용량의 직접사상 캐시와 순차적인 접근이 이루어지는 2-웨이 연관사상 버퍼를 가지므로 동적인 메모리 접근 시간이 가능하기 때문이다.

DM_64K, 4W_32K, VT_32K의 경우, 제안된 캐시 메모리 구조와 달리 항상 접근 시간은 2Cycle로 고정되어 있다. 따라서 비록 캐시 메모리 접근 실패율에 효과적인 성능 향상을 가진다하여도 제안된 캐시 메모리의 동적 접근 시간에 비해 2Cycle 메모리 접근 시간은 전체 시스템의 성능저하의 원인이 된다.

STA_16K는 순차적인 접근으로 제안된 캐시 메모리와 유사한 동작으로 구동된다. 하지만, 그림 3과 그림 4에서 보듯이 STA_16K는 접근 실패율과 평균 메모리 접근 시간에서 제안된 캐시 메모리에 비해 낮은 성능을 보인다. 이는 STA_16K가 비록 순차적인 빠른 접근 시간을 가진다하여도, 2-웨이 연관사상 캐시 메모리의 접근 실패율을 벗어나지 못하기 때문이다. 이에 반해 제안된 캐시 구조는 빠른 접근의 직접사상 캐시와 접근 실패에 효과적인 연관사상 버퍼로 접근 실패에 대한 성능개선을 이룰 수 있었다. 비록, 접근 실패율과 평균 메모리 접근시간이 효과적인 성능향상을 보였다하여도, 오늘날 시스템의 소비전력은 더 이상 묵과 할 수 없는 성능평가의 한 요인으로 자리 잡고 있다. 이에 본 논문에서는 앞서 언급한 것과 같이 소비전력에 관한 평가로 에너지*지연 시간을 측정하였다. 에너지*지연시간은 소비전력과 메모리 접근을 위한 지연시간을 모두 고려한 성능평가 지표라 할 수 있다.

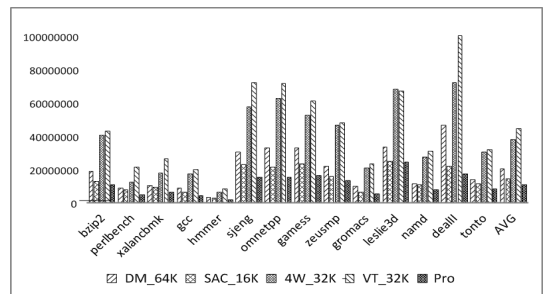


그림 5. 에너지*지연시간(nj*Cycle)
Fig 5. Energy delay product(nj*Cycle)

그림 5는 제안된 구조와 비교 구조 캐시 메모리의 에너지*지연시간(에너지 * 평균 메모리 접근 시간)을 나타낸 그림이다. 그림에서 보듯이 제안된 구조는 비교 구조의 캐시 메모리에 비해 효과적인 성능향상을 보이고 있다. 비록 4W_32K,

VT_32K는 접근 실패율과 평균 메모리 접근 시간에 효과적인 성능을 보이지만, 구조적 특징의 높은 소비전력으로 에너지*지연시간 지표에서는 높은 성능저하를 보이고 있다.

또한 DM_64K는 빠른 접근을 보장하지만, 용량에 비해 높은 접근 실패율과 큰 용량의 소비전력으로 역시 제안된 구조에 비해 높은 에너지*지연시간을 가진다.

STA 역시 동적인 메모리 접근 시간과 제안된 구조와 동일한 크기의 캐시 메모리를 가지지만, 높은 접근실패율과 평균 메모리 접근 시간으로 낮은 소비파워를 가짐에도 제안된 구조에 비해 높은 에너지*지연시간을 보인다.

시뮬레이션 결과 제안된 구조는 DM_64K, 4W_32K, VT_32K에 비해 약 45%, 70% 그리고 75%의 성능향상을 보였다. 또한 STA에 비해 25%의 에너지*지연시간의 성능향상을 보였다.

IV. 결론

본 논문에서는 빠른 클럭 속도를 가지는 CPU에 대해 고성능 내장형 L1 캐시 메모리를 구조를 제안하였다. 제안된 캐시 메모리는 빠른 접근 속도를 위한 직접사상 캐시와 접근 실패율에 효과적인 2-웨이 사상 캐시 메모리로 구성되며, 캐시 메모리는 동적인 접근 시간을 가지므로 기존의 고정된 접근 시간에 비해 효과적인 성능 향상을 이룰 수 있었다.

비교 캐시 메모리의 비해 용량 차이점으로 높은 접근 실패율을 보이지만, 평균 메모리 접근에서는 2배 크기의 4W에 비해 단 6%의 성능저하를 보였으며, 그 외 비교 캐시 메모리에 비해 좋은 성능효과를 보였다. 특히, 에너지*지연시간의 성능 평가에서는 제안된 구조가 모든 비교 캐시 메모리에 비해 좋은 성능향상을 가져왔다. 따라서 제안된 구조는 빠른 클럭 속도를 가지는 CPU에 적합한 저전력, 고성능의 내장형 캐시 메모리 구조라 할 수 있다.

참고문헌

- [1] Intel Processor, <http://www.intel.com/content/www/us/en/homepage.html>
- [2] Samsung DRAM, <http://www.samsung.com/global/business/semiconductor/>
- [3] S. Santhanam, "StrongARM SA110-a 160MHz 32b 0.5W CMOS ARM processor," Hot Chips 8.1996.
- [4] J. H. Lee, J. S. Lee, and S. D. Kim, "A New Cache Architecture Based on Temporal and Spatial Locality," *J. Systems Architecture*, vol.46, pp.1451-1467, Sept., 2000.
- [5] CACTI 4.3, http://www.hpl.hp.com/personal/Norman_Jouppi/cacti4.html
- [6] D. Nicolaescu, A. Veidenbaum, and A. Nicolai, "Using a Way Cache to Improve Performance of Set-Associative Cache," *High-Performance Computing Lecture Notes in Computer Science* Vol.4759, pp.93-104, 2008.
- [7] J. Kang, S. Lee, and I. Lee, "Way-tracking set-associative caches," *Electronics Letters*, Vol. 46, pp.1497-1499, Oct., 2010.
- [8] C. J. Janraj, T. V. kalyan, T. Warriar, and M. Mutyama, "Way Sharing Set Associative Cache Architecture," *25th International Conference on VLSI Design*, pp.251-256, 2012
- [9] C. H. Ting, J. D. Huang, and Y. H. Kao, "Cycle-Time-Aware Sequential Way-Access Set-Associative Cache for Low Energy Consumption," *IEEE Asia Pacific Conference on Circuits and Systems*, pp.854-857, 2008.
- [10] Rolan, D. Fraguera, B. B. Doallo, R. "Adaptive line placement with the Set Balancing Cache," *ACM International Symposium on Microarchitecture*, pp.529-540, 2009.
- [11] Henning, John L., "SPEC CPU2006 benchmark descriptions," *ACM SIGARCH Computer Architecture News* Vol. 34, No. 4, pp. 1-17, Sep., 2006.
- [12] SPEC Benchmark Suite. Information available at <http://www.spec.org/cpu2006/>
- [13] Luk, C., Cohn, R., Muth, R., Patil, H., Klauser, A., Lowney, G., Wallace, S., Vijay Janapa Reddi, and Hallwood, K. Pin: building customized program analysis tools with dynamic instrumentation. In *Proceedings of the 2005 ACM SIGPLAN Conference on Programming Language Design and Implementation*, pp.190-200, June, 2005.
- [14] Intel Pin-tools, <http://software.intel.com/en>

-us/articles/pin-a-dynamic-binary
-instrumentation-tool

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