

논문 2016-11-40

차세대 CPU를 위한 캐시 메모리 시스템 설계 (Design of Cache Memory System for Next Generation CPU)

조 옥 래, 이 정 훈*
(Ok-Rae Jo, Jung-Hoon Lee)

Abstract : In this paper, we propose a high performance L1 cache structure for the high clock CPU. 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 reduce miss ratio, and a way-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 stored into the two-way set associative buffer. For the high performance and fast access time, we propose an one way among two ways set associative buffer is selectively accessed based on the way-select table (WST). According to simulation results, access time can be reduced by about 7% and 40% comparing with a direct cache and Intel i7-6700 with two times more space respectively.

Keywords : High performance CPU, Cache memory, Average memory access time

1. 서 론

오늘날, 컴퓨팅 시스템은 대용량 및 복잡한 데이터의 빠른 처리 속도를 위해 빠른 클럭 속도의 프로세서를 요구하고 있다. 실제, Intel의 i7-6700은 기본 클럭 3.4GHz에서 터보 부스트 기술을 사용하여 최대 4.0GHz까지를 CPU가 보장하고 있다 [1]. 또한 메모리의 집적도 및 접근 속도 역시 급격하게 발전하고 있다 [2]. 비록, 메모리의 접근 속도가 향상되고 있더라도, 여전히 빠른 클럭 속도의 프로세서와 메모리의 속도 차이는 시스템의 성능저하의 원인이 된다.

공정의 기술이 개발되면서 현재 클럭 속도는 급속도로 증가하고 있다. 이러한 빠른 클럭 속도의 프로세서가 제공되어도 여전히 소비전력 및 발열현상은 심각한 문제점을 내포하고 있으며, 또한 빠른 클럭

표 1. 다양한 캐시메모리의
1-Cycle 접근 경계 속도(GHz)

Table 1. The various cache memory boundary speed for 1-Cycle access times(GHz)

	DM	2W	4W
8KByte	4.30	3.45	3.27
16KByte	3.78	2.91	2.89
32KByte	3.52	2.24	2.73
64KByte	3.13	2.12	2.09

속도의 프로세서는 칩 내부 캐시 메모리에 추가적인 접근 시간을 요구하게 된다.

현재 Intel의 i7-6700의 경우 트라이게이트 트랜지스터 기술을 바탕으로 14nm 공정의 고성능 데이터 처리를 위한 빠른 클럭 속도의 프로세서이다. 비록 빠른 데이터 처리를 위해 빠른 클럭 속도의 프로세서가 제공된다 하여도 칩 내부 캐시 메모리에 대한 추가적인 접근 시간을 요구하게 된다. 예로 i7-6700의 경우 L1 캐시 메모리는 32KB의 8-웨이 집합연관 (set associative) 캐시로 구성되어 있다. 이 경우 L1 캐시에 접근할 경우 4-cycle이나 걸리는 단점을 가지고 있다. 따라서 본 논문에서는

*Corresponding Author (leejh@gsnu.ac.kr)

Received: Aug. 4 2016, Revised: Sep. 28 2016, Accepted: Sep. 30 2016.

O.R. Jo: GyeongSang National University

J.H. Lee: GyeongSang National University (ERI)

※ 이 연구는 2015년도 경상대학교 발전기금재단 재원으로 수행되었음

이러한 고성능 프로세서에 적합한 캐시 메모리의 구조를 제안하고자 한다.

본 논문에서는 제안된 구조를 위해 4GHz의 클럭 속도를 가지는 프로세서로 가정하였으며, 이 때 빠른 접근 속도를 고려하여 표 1를 참고하여 선택하였다. 표 1. 현재 가장 기본적으로 사용되는 캐시 메모리의 1-Cycle 접근 경계 속도를 나타낸 표이다 [3, 4].

II. 관련 연구

오늘날, 컴퓨팅 시스템에서 캐시 메모리는 성능 향상 및 저전력을 위해 가장 중요한 역할 뿐 아니라, 그 비중이 점점 커지고 있다. 이에 따라 저전력 및 고성능 캐시 메모리에 대한 다양한 연구가 이루어지고 있다.

4GHz 고성능 CPU 위한 캐시 메모리 시스템 [3]의 연구는 4GHz의 빠른 클럭 속도의 CPU에 적합한 고성능 L1 캐시 메모리 구조로써 버퍼 선택 테이블을 이용하여 2-웨이 집합연관 버퍼중 하나의 웨이만 선택적으로 접근하여 평균 메모리 접근 시간과 소비전력을 향상시킨 연구이다.

WSSA cache [5]는 집합연관 캐시의 특정 블록의 접근 빈도를 고려하여 웨이의 블록을 공유하여 성능을 향상시켰다. Set-Balancing Cache [6]는 빈번하게 접근 되는 블록의 접근 실패 시 자주 사용되지 않는 블록에 데이터를 저장하게 된다. 하지만, WSSA cache와 Set-Balancing Cache는 기존 집합연관 캐시 메모리와 동일한 접근 시간을 가지며, 캐시 메모리 접근 실패 시 공유 블록 혹은 다른 관련 블록의 접근을 위한 추가적인 접근 시간을 요구하게 된다.

SWA [7]는 기존 2-웨이 집합연관 캐시 메모리에서 가장 최근에 접근 한 블록을 순차적으로 접근하여 소비 전력 및 평균 메모리 접근 시간을 향상시켰다. 그러나 기존 2-웨이 집합연관 캐시 메모리와 동일한 접근 실패율을 보이고 있으므로 2-웨이 집합연관 캐시메모리에 한정된 구조라고 할 수 있으며, 웨이가 증가할수록 이 구조는 오히려 성능저하를 야기시킬 수 있다.

III. 본 론

3.1 제안 동기 및 방법

본 논문의 주목적은 급속하게 증가하는 CPU 클

럭 속도에 적합한 캐시 메모리 시스템 설계에 있다.

현재의 컴퓨팅 시스템에서 캐시 메모리는 시스템의 빠른 수행 시간과 성능 향상의 가장 중요한 역할을 수행하고 있다.

캐시 메모리의 성능을 향상시킬 수 있는 방법은 캐시 메모리에서의 접근 실패와 접근 시간을 줄이는 것이다. 접근 실패와 접근 시간을 효과적으로 줄이기 위해 프로그램 수행 시 적합한 두 지역성 (시간적/공간적 지역성)을 많이 이용한다.

공간적 지역성은 최근에 참조된 명령어나 데이터의 이웃이 가까운 미래에 다시 참조될 확률이 높다는 것이며, 시간적 지역성은 최근에 참조된 명령어나 데이터가 가까운 미래에 다시 참조될 확률이 높다는 것을 의미한다. 따라서 공간적 지역성은 블록 크기가 클수록, 시간적 지역성은 블록 개수가 많을수록 더욱 효과적으로 사용할 수 있다. 하지만, 이러한 블록의 크기 및 개수의 증가는 접근 시간 및 소비전력의 증가를 초래할 수 있다.

공정 기술의 발전으로 범용 컴퓨팅 시스템뿐만 아니라 임베디드 시스템의 클럭 속도는 고성능을 위해 더욱 빠른 클럭 속도를 가질 것이다. 일반적으로 접근 실패를 줄이기 위해 캐시 메모리는 집합연관 구조를 사용하고 있다. 이러한 캐시 구조는 메인 메모리에 비해 빠른 접근 시간을 보장하지만, 빠른 클럭 속도를 가지는 CPU에 대해 기존 상용화된 캐시 구조는 추가적인 접근 시간을 요구하게 된다.

본 논문에서는 빠른 CPU 클럭 속도에 대한 성능향상에 효과적인 새로운 캐시 메모리 구조를 제안하고자 한다. 제안된 캐시 구조는 작고 빠른 클럭 속도를 가지는 CPU (4GHz)에 대해 1-cycle을 보장함으로써 성능향상을 높이고자 하였다.

빠른 접근 시간을 위해 주 캐시 메모리는 직접 사상 캐시 (direct-mapped cache) 구조를 선택하였다. 그러나 직접사상 캐시 구조의 단점인 높은 접근 실패율을 줄이기 위하여 희생 (victim) 버퍼를 사용하는 것이 효과적이다. 또한 접근 실패율 관점에서 간단하면서도 효과적인 성능향상을 보이는 구조는 집합연관 구조이다. 하지만 집합연관 구조는 높은 웨이를 가질 때, 소비 전력뿐만 아니라 접근 시간까지 높아지는 단점을 가진다. 따라서 본 논문에서는 2-웨이 집합연관 버퍼를 주 캐시 보조 버퍼로 선택하였다.

제안된 캐시 구조는 접근 실패율을 낮추기 위하여 2-웨이 집합연관 버퍼 구조를 가지지만 빠른 접근을 위하여 하나의 웨이 또는 두 개 또는 아무 웨이 선택 없이 바로 메인 메모리로 접근하고자 하는

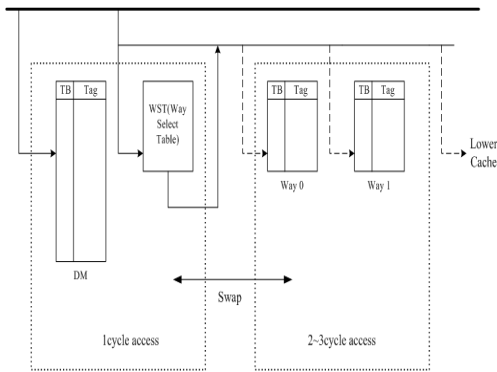


그림 1. 제안된 캐시 구조
Fig. 1 Proposed cache structure

알고리즘을 제안하기 위하여 새로운 웨이 선택 테이블 (WST) 사용하였다.

WST의 접근 시간을 없애기 위하여 주 캐시 메모리인 직접사상 캐시와 WST는 동시에 접근하여 집합연관 버퍼의 웨이를 선택하게 된다. 따라서 제안된 캐시 메모리는 주로 1-cycle을 보장하지만 주 캐시 메모리의 접근 실패에 따라 다양한 2-cycle 또는 3-cycle 접근 시간을 가질 수 있다.

3.2 제안된 캐시 메모리 구조 및 동작

본 논문에서 제안된 캐시 메모리의 구조는 그림 1과 같다. 제안된 캐시 메모리는 빠른 클럭 속도를 가지는 고성능 CPU의 성능향상을 위해 세부분으로 나누어 구성된다. i) 빠른 접근을 위해 직접사상 캐시를 이용한 주 캐시 메모리와 ii) 직접사상 캐시 메모리의 접근 실패율을 줄이기 위한 2-웨이 집합연관 버퍼, 그리고 iii) 버퍼의 정보와 접근 순위를 저장하고 있는 WST (Way Select Table)로 구성된다.

주 캐시 메모리는 CPU의 클럭 속도가 4GHz로 가정했을 때 1-Cycle 접근이 가능한 메모리 크기를 가진다. 또한 주 캐시 메모리는 집합연관 버퍼에서 참조가 일어난 데이터인지 아닌지를 판단하기 위하여 추가적인 1비트 (temporal bit-TB)를 가지고 있다. TB는 2-웨이 집합연관 버퍼에서 접근 성공 후 주 캐시 메모리로 이동할 때 주 캐시 메모리의 해당 TB 비트에 "1"로 갱신되어진다.

2-웨이 집합연관 버퍼는 직접사상 캐시 메모리의 단점인 접근 실패율을 보완하기 위해 사용하였으며 집합연관 버퍼 또한 주 캐시 메모리와 마찬가지로 시간적 지역성을 활용하기 위해 추가적인 1비트인 TB 비트를 가진다.

기본적인 동작은 CPU로부터 데이터가 요청되면, 먼저 주 캐시 메모리와 WST가 동시에 접근이 일어난다. 주 캐시 메모리에서 접근 성공이면 요청된 데이터를 CPU로 전달하게 된다. 만약 접근 실패가 발생하면, WST에 의해 집합연관 버퍼로 접근할지 말지를 선택하게 된다. WST에 의해 선택된 접근 방법에 의해 집합연관 버퍼에 접근이 발생하고, 만약 접근 성공이면, 주 캐시 메모리와 접근 성공이 이루어진 웨이의 데이터는 스왑 (Swap) 동작이 이루어진다. 데이터의 스왑 동작 시 주 캐시 메모리의 TB는 "1"로 갱신되어진다. 웨이 블록으로 스왑 이동하는 주 캐시의 데이터 및 TB는 그대로 이동하게 된다. 또한 WST내부의 LSB, BXOR, LRU 비트도 갱신되어진다. 이에 대한 자세한 설명은 차후에 설명할 것이다.

만약, 집합연관 버퍼에서도 접근 실패가 발생하면, CPU가 요청한 데이터는 하위 계층 메모리에서 주 캐시 메모리로 데이터를 가져오게 되며, 이 때 주 캐시 메모리의 블록의 데이터가 유효하다면, 집합연관 버퍼에 저장하게 된다. 집합연관 버퍼의 희생 블록의 선택은 각 웨이 블록의 TB에 의해 결정하게 된다. 만약 두 웨이 블록의 TB가 동일하면, LRU에 의해 희생 블록을 선택하며, 그렇지 않으면 TB가 "0"인 웨이의 블록을 희생 블록으로 선택하게 된다.

2-웨이 집합연관 버퍼의 접근은 WST에 의해 결정되는데 WST는 가장 최근에 집합연관 버퍼에 저장된 태그 최하위 1비트가 저장된 LSB와 각각의 웨이에 저장된 태그 최하위 1비트들의 XOR 연산의 결과를 저장하는 BXOR (buffer_XOR), 그리고 집합연관 버퍼의 LRU 1비트로 구성된다.

CPU에서 요청된 주소의 태그 최하위 1비트와 LSB를 XOR 연산을 통해 동일한 값을 가지고 있는지 판별하는 동시에 BXOR를 이용하여 표 2와 같이 해당 집합연관 버퍼에 접근을 하게 된다. 표 2에서 \oplus LSB는 요청된 주소의 태그 최하위 1비트와 LSB의 XOR 연산의 결과로 "0"은 요청된 주소의 태그 최하위 1비트와 최근에 집합연관 버퍼에 저장된 태그 최하위 1비트가 동일하다는 것을 의미한다. "1"은 요청된 주소의 태그 최하위 1비트와 최근에 집합연관 버퍼에 저장된 태그 최하위 1비트가 다르다는 것을 의미한다. BXOR 비트는 집합연관 버퍼에 저장된 태그 최하위 1비트들의 XOR 연산의 결과로 "0"은 각각의 웨이에 저장되어 있는 태그 최하위 1비트가 같다는 것을 의미하고 "1"은 다르다는 것을 의미한다.

표 2. WST 동작 알고리즘
Table 2. WST operation algorithm

\oplus LSB	BXOR	
0	0	Sequential access
0	1	Access by LRU
1	0	miss
1	1	Access by !LRU

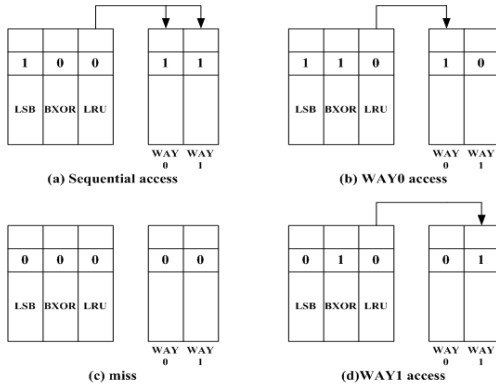


그림 2. 연관 사상 버퍼 접근을 위한 WST 동작 알고리즘

Fig. 2 WST operation algorithm for set-associative buffer access

LSB와 XOR 연산의 결과와 BXOR 비트를 이용하여 '00'일 경우 요청된 주소의 태그 최하위 1비트와 집합연관 버퍼의 각각의 웨이 블록의 태그 최하위 1비트가 같으므로 LRU에 의해 최근에 참조된 웨이 블록부터 순차접근을 하게 된다. 해당 웨이 블록에서 접근 실패가 발생하면 다른 웨이 블록에 접근을 하게 된다. '01'일 경우 요청된 주소의 태그 최하위 1비트와 가장 최근에 참조된 웨이 블록의 태그 최하위 1비트만 같으므로 LRU에 의해 가장 최근에 참조된 웨이 블록만 접근을 하게 된다. '10'일 경우 요청된 주소의 태그 최하위 1비트와 집합연관 버퍼의 각각의 웨이 블록의 태그 최하위 1비트가 다르므로 집합연관 버퍼에 추가적인 접근 없이 접근 실패가 이루어진다. '11'일 경우 요청된 주소의 태그 최하위 1비트와 집합연관 버퍼의 가장 최근에 참조된 웨이 블록의 태그 최하위 1비트는 다르지만 또 다른 웨이 블록의 태그 최하위 1비트는 같으므로 해당 웨이 블록만 접근을 하게 된다.

그림 2는 WST를 이용한 2-웨이 집합연관 버퍼의 접근을 나타낸 그림이다. 만약 CPU가 요청한 주소가 1101001이고, 웨이 버퍼와 WST가 각각 다른

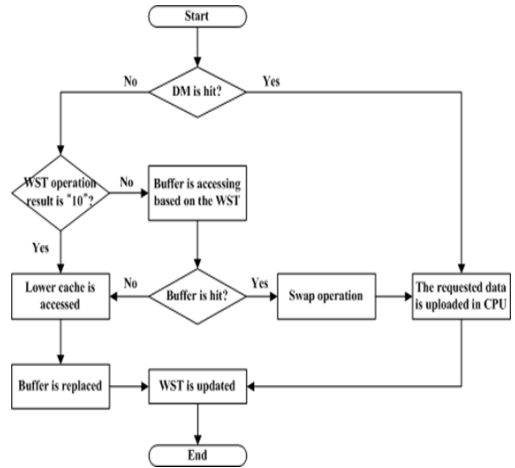


그림 3. 캐시 동작 흐름도
Fig. 3 Flow chart of cache operation

4가지 경우에 대해서 나타낸 그림이다.

직접사상 캐시와 2-웨이 집합연관 버퍼가 동일한 캐시 메모리 크기를 가질 때, 직접사상 캐시 메모리의 태그가 110, 인덱스가 1001 경우, 집합연관 버퍼의 태그는 1101 그리고 001의 인덱스를 가진다. 이 때 집합연관 버퍼의 태그 최하위 1비트("1")를 이용하여 LSB 비트와 XOR 연산을 하고 BXOR를 이용하게 된다. 먼저 (a)에 LSB와 XOR 연산을 하면 $1\oplus 1=0$ 의 결과가 나오게 된다. 그리고 BXOR가 0이므로 "00"이 되고 이 경우 LRU에 의해 웨이 버퍼에 순차적으로 접근하게 된다. (b)는 $1\oplus 1=0$ 의 결과는 동일하지만 BXOR이 1이므로 "01"이 되고 LRU에 의해서 가장 최근에 참조된 WAY 0에 접근을 하게 된다. (c)는 $0\oplus 1=1$, BXOR이 0이므로 "10"이 되고 이 경우 접근 실패가 되며 웨이 접근없이 바로 하위 계층의 메모리에 접근하게 된다. 마지막으로 (d)는 $0\oplus 1=1$, BXOR이 1이므로 "11"이 되고 가장 최근에 참조되지 않은 WAY 1에만 접근을 하게 된다.

그림 3은 본 논문에서 제안된 캐시 메모리의 동작을 흐름도(flow chart)를 이용하여 표현한 것이다. 앞에서 설명한 동작을 간략하게 표현하였다.

3.3 성능평가

본 논문에서 제안된 캐시 구조의 성능평가를 위하여 캐시 메모리의 대표적인 성능 평가 지표인 접근 실패율(miss ratio)와 평균 메모리 접근 시간(Average Memory Access Time)를 이용하였다.

표 3. 시스템 환경

Table 3. System environment

	Value
CPU clock speed	4 GHz
DM cache access time	1 cycle
2-way buffer access time	2~3 cycle
Main memory access time	242 cycle

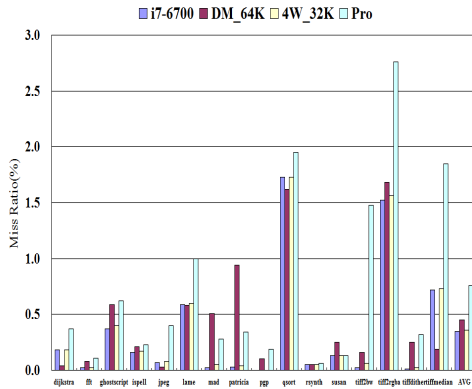


그림 4. 메모리 접근 실패율 (%)
Fig. 4 Memory Miss ratio (%)

표 3은 제안된 캐시 구조의 성능 평가를 위한 시스템 환경 변수를 나타낸 표이다.

제안된 캐시 메모리의 성능평가를 위해 본 논문에서는 Mibench 벤치마크 [8]를 SimpleScalar [9] 시뮬레이터를 이용하여 메모리에 접근하는 트레이스 파일을 추출하여 사용하였다.

제안된 캐시 메모리의 성능 평가를 위해 본 논문에서는 비교 캐시 구조로 가장 상용화된 캐시 구조인 직접 사상 캐시 (DM), 4-웨이 집합연관 캐시 (4W)를 사용하였다. 그리고 Intel i7-6700의 L1 캐시 구조인 8-웨이 집합연관 캐시를 사용하였다. 캐시 메모리의 용량은 DM은 64KB를 사용하였고, 4W와 i7-6700은 32KB, 제안된 캐시 구조 (Pro)는 16KB만을 사용하였다.

그림 4는 캐시 접근 실패율을 나타낸 그림이다. 제안된 캐시 구조가 다른 구조들에 비해 가장 높은 접근 실패율을 보이고 있다. mad, patricia에서는 DM에 비해 낮은 접근 실패율을 보이고 있으나 대체적으로 높은 접근 실패율을 보이고 있다. 평균 접근 실패율을 보면 4W와 i7-6700에 비해 약 2배의

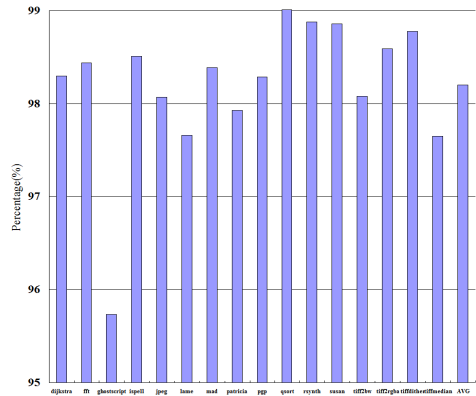


그림 5. 1cycle 접근 성공의 비율 (%)
Fig. 5 Access success ratio of 1cycle

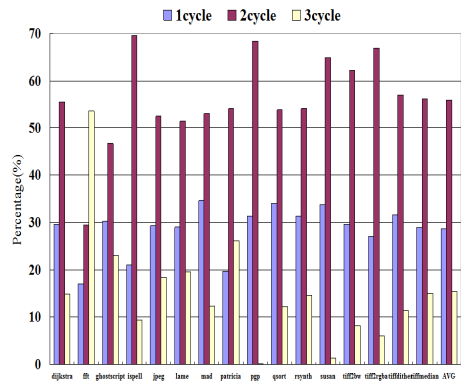


그림 6. 접근 실패시 cycle의 비율
Fig. 6 The percentage of each cycle occurs when access fails

접근 실패율을 보이고 있다. 이는 캐시 메모리의 구조적인 문제를 제외하더라도 2배에서 4배가량의 용량적인 차이가 있기 때문이다.

그림 5는 제안된 캐시 메모리 구조에서 1-cycle 접근 성공의 비율을 나타낸 그림이다. 이는 1-cycle만에 접근 성공을 말하는 것이며, 즉 주 캐시 메모리인 직접 사상 캐시 (8K)에서 접근 성공이 발생한 비율을 말한다. 모든 벤치마크에서는 95%이상 1-cycle 접근에 성공하였고, 평균 98%이상 1-cycle 접근에 성공하였다. 다른 캐시 메모리 구조와 달리 제안된 캐시 메모리가 1-cycle에 접근이 가능한 8KB의 작은 직접 사상 캐시 메모리를 사용하였기 때문이며, 이는 평균 메모리 접근 시간에서 더 효과적인 모습을 보이고 있다.

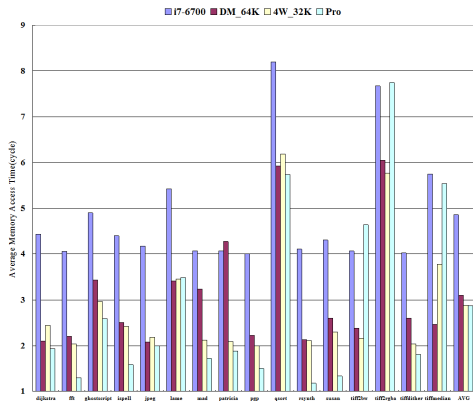


그림 7. 평균 메모리 접근 시간 (cycle)

Fig. 7 Average Memory Access Time (cycle)

그림 6은 제안된 캐시 메모리에서 접근 실패가 발생했을 때 각 cycle의 비율을 나타낸 그림이다.

1-cycle 접근 실패는 fft, ispell, patricia를 제외한 나머지 벤치마크에서는 약 30%정도이며, 2-cycle 접근 실패는 fft, ghostscript를 제외한 나머지 벤치마크에서는 50%이상이다. 평균적으로 1-cycle 접근 실패는 약 28%, 2-cycle 접근 실패는 약 56%, 3-cycle 접근 실패는 약 15%를 차지하고 있다.

그림 7은 평균 메모리 접근 시간을 나타낸 그림이다. 다른 캐시 메모리 구조는 표 1를 이용하여 i7-6700은 접근 시간을 4-cycle, 나머지 캐시 메모리 구조는 항상 접근 시간을 2-cycle로 고정하여 시뮬레이션을 수행하였다. 시뮬레이션 결과 제안된 구조는 tiff2bw, tiff2rgba, tiffmedian를 제외한 나머지 벤치마크에서 평균 메모리 접근 시간의 성능이 가장 좋은 것을 확인할 수 있다. 비록 제안된 캐시 메모리의 구조가 4W와 i7-6700보다 접근 실패율에서는 약 2배의 성능 저하를 보이고 있지만 오히려 평균 메모리 접근 시간에서는 i7-6700에 비해 평균 약 40%정도 성능 향상을 보이고 있으며, 4W와 비교해봤을 때 거의 동일한 모습을 보이고 있다.

그러므로 클럭 속도가 증가할수록 1-cycle 접근이 어려운 큰 용량의 캐시 메모리는 접근 실패율에서는 효과적이지만 성능향상에는 도움이 되지 못함을 알 수 있다. 또한 부수적으로 소비 전력면에서 캐시 용량이 클수록 접근하는 캐시 소비전력이 증가하기 때문에 제안된 구조의 소비전력은 매우 줄어들 것으로 판단된다.

IV. 결론

본 논문에서는 빠른 클럭 속도 (4GHz)를 가지는 CPU에 대해 고성능 내장형 L1 캐시 메모리의 구조를 제안하였다. 제안된 캐시 메모리는 빠른 접근 속도를 위해 1cycle 접근이 가능한 직접사상 주 캐시 메모리와 주 캐시 메모리의 단점인 접근 실패율에 효과적인 2-웨이 연관 사상 버퍼로 구성되어 있다. 또한 제안된 캐시 메모리는 동적인 접근 시간을 가지므로 기존의 고정된 접근 시간에 비해 효과적인 성능 향상을 이룰 수 있다.

비교 캐시 메모리와 용량 차이로 인하여 더 높은 약 2배의 접근 실패율을 보이지만, 평균 메모리 접근에서는 오늘날 사용되고 있는 i7-6700보다 약 40%이상의 성능향상을 보이고 있다. 그리고 다른 비교 캐시 메모리 구조와 비교했을 때 더 효율적인 성능향상을 보였다. 따라서 제안된 캐시 메모리의 구조는 빠른 클럭 속도를 가지는 CPU에 적합한 고성능의 내장형 캐시 메모리 구조라고 할 수 있다.

본 논문에서는 Mibench 벤치마크를 이용하여 시뮬레이션을 진행하였다. 향후 주로 사용되고 있는 SPEC2006 등 다양한 벤치마크에서도 시뮬레이션을 진행하여 개선점이 있는지 확인하도록 하겠다.

References

- [1] Intel Processor, <http://www.intel.com/content/www/us/en/homepage.html>
- [2] Samsung DRAM, <http://www.samsung.com/semiconductor/>
- [3] B.S. Jung, J.H. Lee, "Cache memory system for high performance CPU with 4GHz," Journal of the Korea Society of Computer and Information, Vol. 18, pp. 1-8, 2013 (in Korean).
- [4] CACTI 4.0, <http://www.hpl.hp.com/techreports/2006/HPL-2006-86.pdf>
- [5] C.J. Janraj, T.V. kalyan, T. Warriar, M. Mutya, "Way sharing set associative cache architecture," Proceedings of IEEE International Conference on VLSI Design, pp. 251-256, 2012.
- [6] D. Rolan, B. Fraguera, R. Doallo, "Adaptive line placement with the set balancing cache," Proceedings of the 42nd Annual IEEE/ACM

International Symposium on Microarchitecture, pp. 529-540, 2009.

- [7] C.H. Ting, J.D. Huang, Y.H. Kao, "Cycle-time-aware sequential way-access set-associative cache for low energy consumption," Proceedings of IEEE Asia Pacific Conference on Circuits and Systems, pp. 854-857, 2008.
- [8] M.R. Guthaus, J.S. Ringenber, D. Ernst, T.M.

Austin, T. Mudge, R.B. Brown, "MiBench: a free, commercially representative embedded benchmark suite," Proceedings of IEEE International Workshop on Workload Characterization, pp. 3 - 14, 2001.

- [9] D. Burger, T.M. Austin, "The simplescalar tool set, version 2.0," ACM SIGARCH Computer Architecture News, Vol. 25, pp. 13-25, 1997.

Ok-Rae Jo (조 옥 래)



He received B.S. degree from Gyeong-Sang National University in 2015. He is currently a M.S. student in Control and Instrument Engineering at Gyeong-

Sang National University. His research interests include cache memory and high performance CPU technologies.

Email: love_jo1210@naver.com

Jung-Hoon Lee (이 정 훈)



He received the M.S. and Ph.D. degree in Computer Science from Yonsei University, Seoul, Korea, in 2001 and 2004, respectively.

He is currently an associate professor in ERI, Control Instrumentation Engineering, GyeongSang National University. His research interests include advanced computer architectures, intelligent memory system, next flash memory, and low power technologies.

Email: leejh@gsnu.ac.kr