모드 선택 비트를 활용한 필터 캐시 예측 모델

곽종욱*, 최주희**, 장성태***, 전주식****
*영남대학교 전자정보공학부
**삼성 전자 SOC 연구소 Proce. Arch. Lab.
***수원대학교 컴퓨터학과, ****서울대학교 컴퓨터공학부
e-mail:kwak@ynu.ac.kr

Filter Cache Predictor using Mode Selection Bit

Jong Wook Kwak*, Ju Hee Choi**, Seong Tae Jhang***, Chu Shik Jhon****
*Dept. of Computer Engineering, School of EECS, Yeoungnam University
**Proce. Arch. Lab. SOC R&D Center, Samsung Electronics
***Department of Computer, The University of Suwon
****School of Computer Science and Engineering, Seoul National Univerisity

요 약

캐시 에너지의 소비 전력을 줄이기 위해 필터 캐시가 제안되었다. 필터 캐시의 사용으로 인해 많은 전력 사용 감소 효과를 가져왔으나, 상대적으로 시스템 성능도 더불어 감소하게 되었다. 필터 캐시의 사용으로 인한 성능 감소를 최소화하기 위해서, 본 논문에서는 기존에 제안된 주요 필터 캐시 예측 모델들을 소개하며, 각각의 방식에 있어서의 핵심 특징 및 해당 방식의 문제점을 분석한다. 이를 바탕으로 본 논문에서는 모드 선택 비트를 활용하는 개선된 형태의 새로운 필터 캐시 예측기 모델을 제안한다. 제안된 방식은 MSB라 불리는 참조 비트를 고안하여, 이를 기존의 필터캐시와 BTB에 새롭게 활용한다. 실험 결과, 제안된 방식은 기존 방식 대비, 전력 소모량 • 시간 지연면에서 평균 5%의 성능 향상을 가져 왔다.

1. 서론

프로그램의 전체적인 실행 속도를 높이거나 프로그램의 실행에 사용되는 전력 소모량을 줄이는 기술들은 많이 개발되어 왔다. 그러나 많은 경우, 실행 속도를 높이면 전력 소모량이 늘어나거나, 전력 소모량을 줄이면 실행 속도가 낮아지게 된다. 필터 캐시(Filter Cache)는 실행 속도 향상과 전력 소모량 감소라는 두 가지 목표를 동시에 추구하는 대신 상호절충(Trade-off)적인 상황을 받아들이고, 적절하게 조화를 이루는 것을 선택하였다[1]. 그래서 필터 캐시는 그 연구 목표를, 일반적인 프로그램에서는 실행 속도의 손실을 보더라도, 내장형 시스템에서 자주 사용되는 멀티미디어나 통신 알고리즘을 최적화하는데 초점을 두었다. 본 논문에서는 기존의 필터 캐시의 발전 과정을 살펴보고, 필터 캐시 연구에서 가장 많이 쓰이는 필터 캐시 예측기 모델(Filter Cache Predictor Model)을 분석해 본다. 이 분석을 바탕으로 기존 방식의 문제점을 해결하는 효율적인 필터 캐시 예측기를 제안한 뒤, 모의실험을 통해 성능의 향상을 알아본다. 이하, 본 논문의 구성은 다음과 같다. 2장에서는 기존의 필터 캐시의 연구에 대해서 알아보고, 기존의 필터 캐시 예측기 모델에 대해 소개한다. 3장에서는 모드 선택 비트를 이용한 예측기 모델(Predictor Model using Mode Selection Bit)을 소개하고, 그 구조 및 동작 방식에 대해서 논의한다. 4장에서는 모의실험을 수행하고, 실험 결과를 통해서 기존의 기법들과 비교, 평가한다. 마지막으로 5장에서는 결론을 내린다.

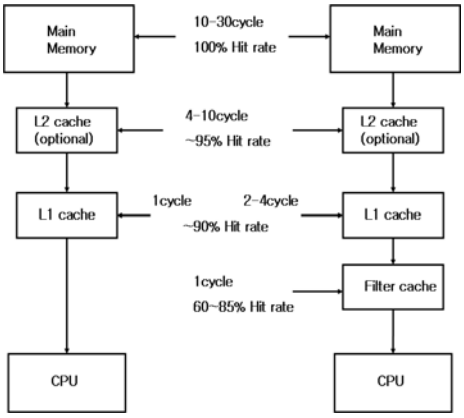
2. 배경 지식 및 관련 연구

표 1에서 보이는 바와 같이, StrongARM 110의 경우, 프로세서 전력의 43%를 캐시 시스템이 사용하고 있다[2]. 캐시 시스템이 전력을 많이 소모한다는 것은 곧 전력 소모를 줄일 수 있는 여지 또한 많이 가지고 있다는 뜻이므로, 캐시 시스템은 내장형 프로세서를 위한 저전력분야에서 중요한 연구의 대상이 되어 왔다.

<표 1> StrongARM110에서의 전력소모

Instruction Cache	27%
Decode	18%
Data Cache	16%
Clock	10%
Execution	8%
others: MMU, Write Buffer, etc	21%

필터 캐시는 그림 1에서 나와 있듯이 기존의 캐시 시스템과 매우 비슷한 구조를 가진다. 기존의 구조에서는 프로세서가 레벨 1 캐시에 직접 접근하지만, 필터 캐시에서는 프로세서가 레벨 1 캐시에 접근하는 대신에 필터 캐시에 접근하게 된다. 만약 프로세서가 인출(fetch)하려는 주소의 명령어가 필터 캐시에 있는 경우에는 필터 캐시에서 명령어를 제공하게 된다. 반대로, 필터 캐시에 존재하지 않는 경우는 레벨 1 캐시로 접근하여 해당 명령어를 가져오고, 다시 그 명령어를 프로세서로 전송하게 된다.

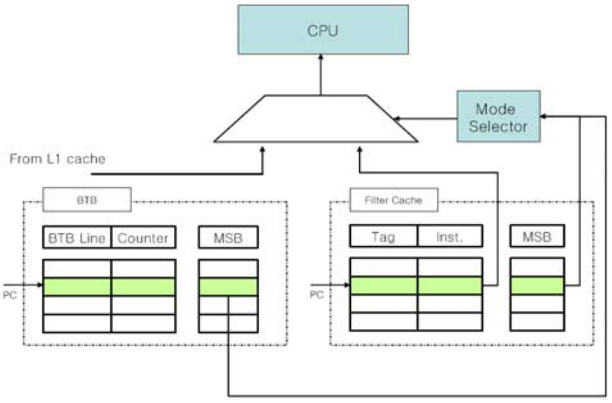


(그림 1) 필터 캐시의 구조

필터 캐시는 그 적중률(hit rate)에 비례해서 시스템의 전력 소모량 감소 효과는 커지게 되며, 필터 캐시의 실패율(miss rate)에 비례해서 시스템의 실행 속도가 저하된다. 그러므로 필터 캐시의 적중률이 높을수록 전반적으로 전력 소모량이 감소하고, 실패율이 높을수록 실행 속도가 저하되는 결과를 나타내게 된다. 필터 캐시는 일반적으로 기존의 캐시 시스템과 비교하여 전력 소모량은 감소하지만, 실행 속도가 떨어지기 때문에 필터 캐시의 성능을 평가하기 위해서는 전력 소모량과 시간 지연을 모두 고려하는 전력 소모량과 시간 지연의 곱(Energy • Delay)을 구해야한다.

3. 필터 캐시 예측 모델

모드 선택 비트를 사용한 필터 캐시 예측기 모델(Filter Cache Predictor Model using Mode Selection Bit)은 관련 연구에서 소개된 2가지 방식을 고려해서 고안되었다. 그림 2에서 보는 바와 같이, 프로그램이 처음 시작하고 필터 캐시가 초기화 되면, 모드 비트는 모두 리셋된다. 프로그램이 시작될 때는 필터 캐시가 비어있으므로 레벨 1 캐시 모드로 동작한다. 프로그램이 진행되다가 현재 인출되는 명령어가 해당 라인의 마지막 명령어라고 판단되는 경우에는 필터 캐시의 모드 비트를 참조하게 된다. 이때, 모드 비트가 리셋되어 있으면, 이 라인은 처음으로 필터 캐시에 들어온 라인이므로, 다음 라인도 역시 필터 캐시에 없을 가능성이 크다. 따라서 레벨 1 캐시 모드로 동작하도록 한 후, 해당 라인은 마지막 명령어까지 실행되었으므로 모드 비트를 세팅시킨다. 반면, 라인의 마지막 명령어이면서 모드 비트가 세팅되어 있으면, 이 라인은 이전에 이미 한 번 실행되었던 라인이며 다음 라인도 역시 필터 캐시에 있을 가능성이 크므로, 모드 선택기가 다음 명령어를 필터 캐시에서 인출하도록 필터 캐시 모드로 모드를 변환하게 한다. 명령어의 주소가 BTB에서 읽히는 경우, 현재 명령어가 분기 명령어라고 판단하고, BTB의 모드 비트를 보고 모드 선택기가 모드를 선택하게 한다.



(그림 2) 필터 캐시 예측기 모델의 구조

그림 2에서 소개된 모델의 각 구성 요소들은 다음과 같은 동작을 수행한다.

• Mode Selector

제안된 모델은 필터 캐시 모드와 레벨 1 캐시 모드 중에 하나의 모드로 동작하게 된다. 필터 캐시 모드(1)는 다음 명령어를 필터 캐시에서 인출하게 되고, 레벨 1 모드(0)는 다음 명령어를 레벨 1 명령어 캐시에서 인출하게 된다. 각각의 모드는 현재 라인이 필터 캐시에 최초로 저장된 라인(모드 0)인지, 이전에 라인 끝까지 실행된 적이 있는 라인(모드 1)인지를 구별한다. 필터 캐시 모드일 때는 필터 캐시와 동일한 방법으로 동작하지만, 레벨 1 캐시 모드일 때는 레벨 1 캐시로 직접 접근하기 때문에, 필터 캐시는 접근하지 않는다.

• Mode Selection Bit(MSB) Table in Filter Cache

필터 캐시의 엔트리 수와 동일한 수의 엔트리를 가지는 1비트짜리 테이블이 필터 캐시 라인에 추가된다. 모드 선택 비트는 필터 캐시에 라인이 저장될 때 리셋(0)되며, 한 번 라인의 마지막까지 실행이 되면 비트가 세팅(1)된다. 라인의 마지막 명령어(word)에 접근할 때 해당 엔트리의 비트를 참조하여 모드 선택기가 모드를 선택한다.

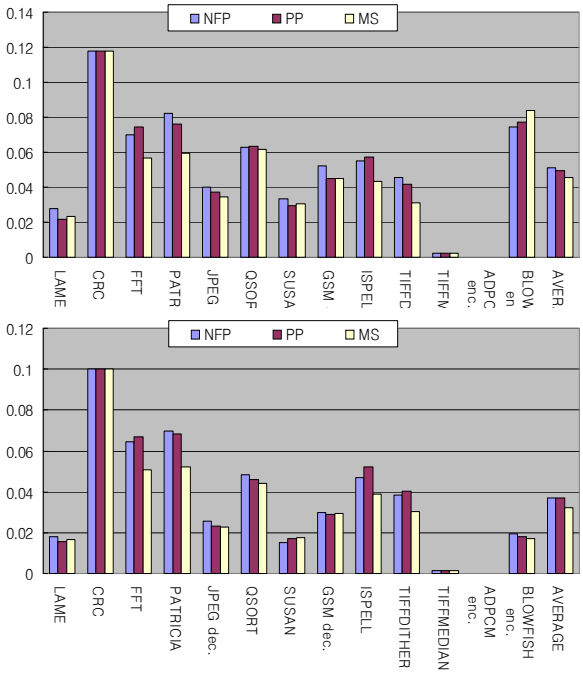
• MSB Table and Counter in BTB

분기 명령어일 때는 BTB의 모드 선택 비트를 참조하여 모드를 선택하게 된다. BTB의 모드 선택 비트는 함께 붙어있는 카운터에 의해 관리된다. BTB의 분기 명령어가 적중하게 되면 카운터 값이 1씩 증가하다가 카운터 값이 임계값(Threshold)에 이르게 되면 라인의 모드 비트는 세팅(1)된다. BTB가 새로운 분기 명령어와 타겟을 라인에 저장하게 되면 모드 선택 비트는 리셋(2)된다.

4. 모의실험 및 성능 분석

내장형 프로세서를 대상으로 하는 본 논문의 검증을 위해서, 슈퍼스칼라 프로세서의 사이클 수준 시뮬레이터인 SimpleScalar ARM 버전을 사용하여 모의실험을 실시하

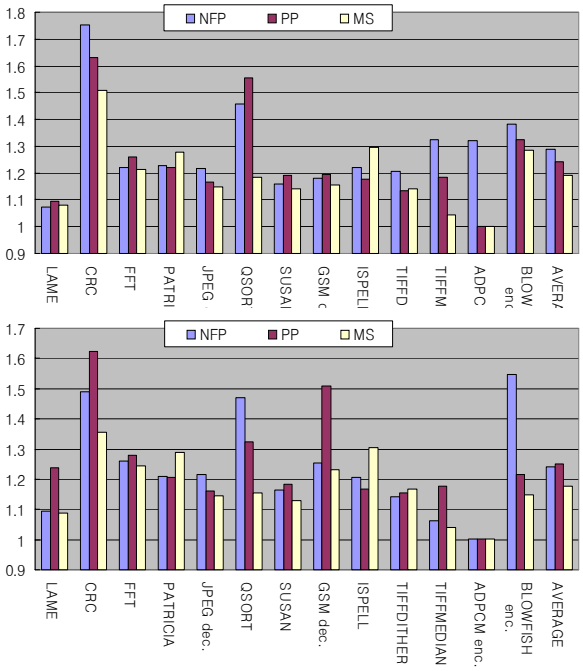
였다[3]. 또한, 전력소모량의 측정을 위해서 CACTI 3.0을 사용하였으며, 0.18um 기술을 가정하였다[4]. 실험대상은 NFP 기법과 패턴 예측 기법(PP)이다. 본 논문에서 제안된 기법은 MS라고 약칭한다.



(그림 3) 예측 실패율

그림 3은 각 예측기의 예측 실패율을 나타낸 것이다. 예측 실패란 예측기가 다음에 인출될 명령어가 필터 캐시에 있다고 예측하였으나, 실제로는 필터 캐시에 없었던 경우를 가리킨다. 이 비율이 높을수록 필터 캐시에서 레벨 1 캐시로 명령어가 포함된 라인을 요청하고 다시 저장하는 시간이 발생하여 시간지연이 발생하게 된다. 그림 3에서처럼, 256B 필터 캐시인 경우에는 NFP가 예측 실패율이 가장 높고, PP가 다음이며 MS가 가장 낮다. 또한, 필터 캐시의 크기가 증가할수록 PP가 NFP보다 예측 실패율이 커지는 것을 볼 수 있다.

그림 4는 각 예측기의 전력 소모량 • 시간 지연을 나타낸 것이다. 본 논문에서 제안한 MS기법이 다른 방식보다 좋은 성능을 보이고 있다. 평균적으로 256B 필터 캐시일 때는 각각 8.1%, 4.2% 감소하였으며, 512B 필터 캐시일 때는 5.4%, 6.1% 감소하였다. 모드 선택 비트 기법은 필터 캐시의 용량이 증가함에 따라 꾸준하게 성능이 향상되는 경향성을 유지하고 있다. 이와 함께, 분기 명령어를 고려한 예측 기법을 사용하고 있어서, 패턴 예측 기법의 장점인 분기 명령어에서 쓰이는 기법들을 필터 캐시 시스템에 접목시킬 수 있는 유연성도 가지고 있다.



(그림 4) 전력 소모량 • 시간 지연

5. 결론

본 논문에서는 모드 선택 비트를 이용한 필터 캐시 예측 시스템을 제안하였다. 제안된 새로운 시스템은, 필터 캐시에 모드 선택 비트를 추가하고, BTB에는 포화 카운터와 모드 선택 비트를 추가한다. 이를 활용하여, 현재 인출되는 명령어가 라인의 마지막 명령어이거나 분기 명령어인 경우, 해당 라인의 모드 선택 비트를 참조하여 다음 명령어의 위치를 예측한다. 모의실험을 통해서 검증한 결과 본 논문에서 제안된 기법은, 기존의 NFP 기법이나 패턴 예측 기법보다 좋은 성능을 보이고 있다. 평균적으로 256B 필터 캐시일 때는 전력 소모량 • 시간 지연이 각각 8.1%, 4.2% 감소하였으며, 512B 필터 캐시일 때는 5.4%, 6.1% 감소하였다.

참고문헌

[1] J. Kin, et al., “The filter cache: An energy efficient memory structure”, In Proceedings of 30th Annual International Symposium on Microarchitecture, December 1997.

[2] J. Montanaro et al. “A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor”, *IEEE Journal of Solid-State Circuits*, 32 (11) : 1703-14, 1996.

[3] D. Burger, et al., “Memory hierarchy extensions to SimpleScalar 3.0”, Technical Report TR99-25, Department of Computer Science, University of Texas at Austin, April 1999.

[4] Shivakumar, P., and Jouppi, N. “An integrated cache timing, power and area model”, Tech. Report, Compaq Western Research Lab, Palo Alto, CA, 2001/2