

분기 동시 수행을 이용한 단일 칩 멀티프로세서의 성능 향상 기법

*이승렬, 정진하, 최재혁, 최상방
인하대학교 전자공학과

e-mail : ziztux@gmail.com, sangbang@inha.ac.kr

Performance improvement of single chip multiprocessor
using concurrent branch execution

*Seung-Ryul Lee, Jin-Ha Jung, Jae-Hyeok Choi, Sang-Bang Choi
Department of Electronic Engineering
Inha University

Abstract

Exploiting the instruction level parallelism encountered with the limit. Single chip multiprocessor was introduced to overcome the limit of traditional processor using the instruction level parallelism. Also, a branch miss prediction is one of the causes that reduce the processor performance. In order to overcome the problems, in this paper, we make single chip multiprocessor having the idle core execute the two control flow of conditional branch. This scheme is a kind of multi-path execution technique based on single chip multiprocessor architecture.

I. 서론

전통적으로 프로세서 성능 향상에 이용되어 오던 명령어 수준의 병렬성은 그 한계를 드러내고 있다[1]. 프로세서의 성능을 저하시키는 가장 큰 요인 중에 하나는 분기 예측 실패이다. 이에 대한 대안으로 연구된 것이 쓰레드 수준의 병렬성을 이용하는 아키텍처들이다. 단일 칩 멀티프로세서(Single chip multiprocessor)는 쓰레드 수준의 병렬성을 이용하는 아키텍처 중 하나이다[2]. 단일 칩 멀티프로세서는 멀티 쓰레드로 작

성된 프로그램들에 대해서는 많은 성능 향상을 보이지만 그렇지 않은 경우에는 최대의 성능을 내지 못한다.

본 논문에서는 이러한 두 가지 문제점을 극복하기 위해 일종의 다중 경로 실행(multi-path execution) 기법[3]을 단일 칩 멀티프로세서 아키텍처에 적용한다. 제안된 구조의 장점은 낭비되는 프로세서를 이용함으로써 프로세서 자원의 효율이 높아진다는 점과 분기 예측 실패가 발생하지 않게 함으로써 분기 예측 실패로 인한 성능 저하를 막을 수 있다는 점이다. 본 논문에서는 핵심이 되는 아이디어와 전체적인 동작에 초점을 맞추어서 기술한다.

II. 본론

단일 칩 멀티프로세서는 여러 개의 완전한 프로세서를 하나의 칩에 집적시킨 프로세서를 말한다. 본 논문에서는 포함되는 여러 개의 프로세서 중 하나를 지칭하는 용어로 '코어'라는 용어를 사용하겠다.

2.1 분기 동시 수행

분기 동시 수행은 단일 칩 멀티프로세서의 코어 중 유휴 상태인 코어를 이용하여 조건 분기의 가능한 두 프로그램 흐름을 모두 수행한다. 각각의 명령어를 수행하는 중에 분기의 방향이 결정되면 실제 분기 방향을 수행 중인 코어는 계속 수행을 진행하고 잘못된 방향을 수행 중인 코어는 수행을 중지하게 된다.

유티 상태에 있는 코어는 이전 시점에 다른 프로그램 명령어를 실행했을 수 있기 때문에 두 코어의 레지스터 파일과 1차 캐쉬의 내용이 일치하지 않는다. 두 코어에서 정상적으로 프로그램이 실행되기 위해서는 적어도 레지스터 파일의 복제가 필요하다. 1차 캐쉬는 크기가 크기 때문에 복제를 하지 않고 코어가 유티 상태로 들어가는 순간부터 수행 중인 코어와 같은 주소를 참조하게 함으로써 선인출(prefetch)과 같은 방법으로 1차 캐쉬를 일치시켜 나간다. 유티 기간이 짧아 1차 캐쉬가 부분적으로만 일치되더라도 2차 캐쉬가 1차 캐쉬의 내용을 모두 담고 있기 때문에 최소한 2차 캐쉬 적중은 보장이 된다.

2.2 분기 동시 수행을 위한 하드웨어 구성

분기 동시 수행을 단일 칩 멀티프로세서에 적용하기 위해서는 단일 칩 멀티프로세서 구조에 약간의 변경이 필요하다. 그림 1은 수정할 프로세서의 구조이다.

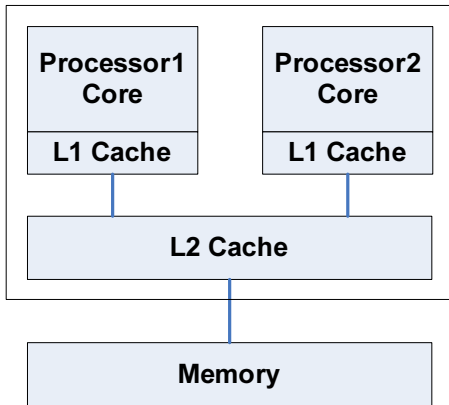


그림 1. 사용되는 단일 칩 멀티프로세서 구조

레지스터 파일의 복제를 위한 통신 채널이 두 코어 간에 필요하다. 다음으로 필요한 것은 1차 캐쉬의 선인출을 위한 각 프로세서의 1차 캐쉬와 2차 캐쉬간의 주소 버스를 멀티플렉싱하는 수정이 필요하다. 또한 현재 코어들의 상태(유티, 분기수행, 스레드 실행)를 가지는 테이블(PST)이 필요하다. 그림 2에 수정된 프로세서의 구조를 간략히 나타내었다.

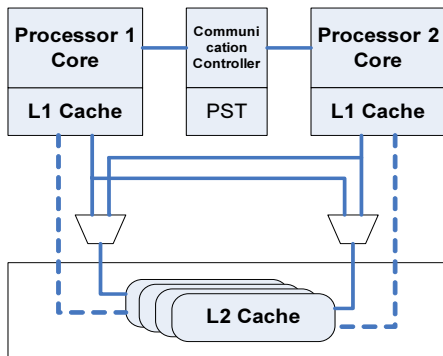


그림 2. 수정된 프로세서 구조

III. 시뮬레이션

제안된 구조를 평가하기 위해 SimpleScalar 프로세서 시뮬레이터와 SPEC2000 벤치마크 프로그램을 이용하여 시뮬레이션을 수행하였다. 그림 3은 단일 스레드 벤치마크 프로그램에 대해서 슈퍼스칼라 프로세서를 기준으로 일반적인 단일 칩 멀티프로세서와 분기 동시 수행 기법을 적용한 단일 칩 멀티프로세서간의 성능을 비교한 것이다.

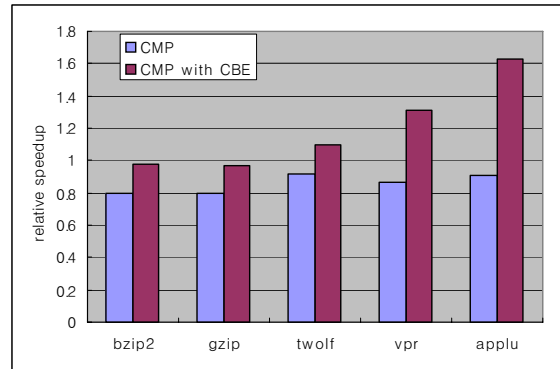


그림 3. 시뮬레이션 결과

IV. 결론 및 향후 연구 방향

본 논문은 기존의 단일 칩 멀티프로세서에서 낭비되는 자원을 이용하고 분기 예측 실패시 발생하는 손실을 줄이고자 분기 동시 수행 기법을 제안하였다. 기존의 일반적인 단일 칩 멀티프로세서에 비해 대략 30%의 성능 향상이 있었으며 2차 캐쉬 실패율이 높은 프로그램에서 더욱 큰 성능 향상이 있었다.

앞으로의 연구는 좀 더 많은 코어를 내장하는 구조에서 효율적인 통신 채널의 구조에 대해 이루어져야 하겠다. 또한 중첩된 분기의 경우에 어느 수준까지 분기 동시 수행을 중첩하여 허용할 것인가에 대한 추가적인 연구도 필요하다.

참고문헌

- [1] J. Hennessy, and D. Patterson, Computer Architecture : A quantitative Approach, 3rd edition, Morgan Kaufmann, pp.172-209, 2002.
- [2] K. Olukotun, and A. Basem, "The case for a single-chip multiprocessor," Proc. 7th Int. symp. Architectural Support for Programming Languages and Operating Systems, October 1996.
- [3] S. Wallace, B. Calder, and D. M. Tullsen, "Threaded multiple path execution," Proc. 25th Int. symp. Computer Architecture, pp. 238-249, July 1998.