슈퍼스칼라 프로세서에서의 값 예측의 전력 소모 측정 및 분석

이명근, 이상정
순천향대학교 정보기술공학부
lmk77@n-top.com, sjlee@sch.ac.kr

An Analysis of Power Dissipation of Value Prediction in Superscalar Processors

Myoung-Kun Lee, Sang-Jeong Lee
Division of Information Technology Engineering, Soonchunhyang University

요 약

고성능 슈퍼스칼라 프로세서에서는 명령어 수준 병렬성(Instruction Level Parallelism, ILP)의 장애인 명령어간의 종속 관계 중 데이터 종속관계를 극복하기 위해 값 예측기를 이용하여 모험적으로 명령어들을 실행한다. 값 예측 시에 필요한 테이블 참조와 값 예측 실패 시 실행되는 잘못된 명령어의 실행은 프로세서의 부가적인 전력 소모를 요구한다. 본 논문에서는 값 예측기와 Cai-Lim 의 전력모델을 슈퍼스칼라 프로세서 사이클 수준 시뮬레이터인 SimpleScalar 3.0 툴셋에 삽입하여 전력 소모량을 측정하고 분석한다.

1. 서론

최근의 고성능 슈퍼스칼라 프로세서에서 높은 성능에 도달하기 위해서는 명령어 수준 병렬성을 이용하여 다수의 명령을 동시에 이슈하고 처리해야 한다. ILP 를 이용하는 주요장애는 데이터종속(Data Dependences) 관계이다. 데이터 종속관계는 현재 명령이 이전 명령의 수행 결과를 참조할 때 발생하고, 이전 명령의 결과가 생성될 때까지 현재의 명령은 실행할 수가 없다. 데이터종속 관계는 와이드 이슈 프로세서에서 명령들 간에 빈번히 발생하고, 명령어 수준 병렬처리의 주요 장애가 되어 고성능 슈퍼스칼라 프로세서의 성능 저하의 주된 요인이 되고 있다. 따라서 최근에는 실행되는 명령의 결과 값을 미리 예측하고, 이후 데이터종속 관계가 있는 명령들에게 값을 조기에 공급하고 이들 명령을 모험적으로 실행하여 성능 향상을 꾀하는 값 예측(Value Prediction) 방식에 관한 활발한 연구가 진행되고 있다[3,4]. 모험적 실행의 성격 상 예측이 실패할 경우 잘못된 명령들의 부가적인 실행으로 인해 전력 소모가 발생하고, 예측에 필요한 부가적인 테이블 참조로 전력 소모량이 증가한다[5].

본 논문에서는 값 예측기와 Cai-Lim 의 전력 모델[2]을 슈퍼스칼라 프로세서 사이클 수준 시뮬에이터인 SimpleScalar 3.0 툴셋[1]에 삽입하여 전력 소모량을 측정하고 분석한다.

2. 값 예측기

모험적 실행 방식 중 값을 미리 예측 하는 방법에는 최근 값 예측(Last Value Predictor)방법 , 스트라이드 값 예측(Stride Value Predictor) 방법, 2-단계 값 예측(2-level Value Predictor) 방법 그리고 혼합형 값 예측(Hybrid Value Predictor)방법이 있다.

최근 값 예측기는 최근에 일어난 값을 근거로 한 명령의 결과 값을 예측하는 기법으로 고정된 상수형 값을 생성하는 명령의 결과를 예측하는데 적합하다[3] 스트라이드 값 예측기는 최근 값 예측기를 확장하여 명령의 결과값이 상수형 값 뿐 만 아니라 고정된 값(스트라이드)으로 변하는 값을 예측하는 방법을 사용한다. 즉, 한 명령에 대한 최근 수행된 2 개의 결과 값의 차인 스트라이드를 구하고 다음에 이 명령의 수행 시 이전 결과 값과 최근 값을 더하여 값을 예측하는

본 연구는 한국과학재단 목적기초연구(2000-1-30300-008-3) 지원으로 수행되었음.

기법이다. 2-단계 값 예측기는 일정한 패턴으로 변하는 값들을 예측하기 위해 명령의 수행결과를 생성된 일련의 결과 값들을 테이블에 적합하고 저장된 패턴을 기준으로 다음과 생성될 값을 예측한다.

앞의 예측기들은 특정 패턴의 명령들에 대해서는 잘 동작하지만 다른 패턴에 대해서는 성능이 저하된다. 본 논문에서는 이들 예측기를 스트라이드와 2-단계 예측기를 통합한 혼합형 값 예측기를 사용한다[4].

3. 전력 소모 측정 도구

본 논문에서는 Cai-Lim 의 전력모델을 기존의 대표적인 슈퍼스칼라 프로세서의 사이클 수준 시뮬레이터인 Simplescalar 3.0 과 결합하여 프로세서의 기능 수준 동작과 프로세서의 전체 전력소모 및 사이클 당 전력 소모를 측정하였다. 그림 1 은 전력 소모 측정 도구의 구성도이다.

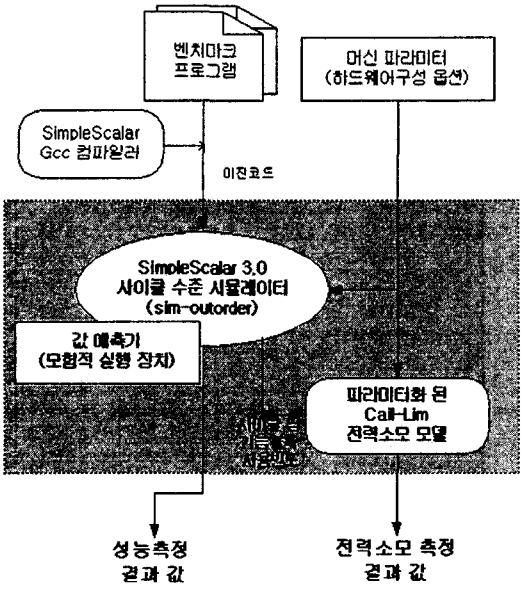

그림 1. 전력소모 측정 도구 구성도

Simplescalar 3.0 툴 셋은 Wisconsin 대학의 슈퍼스칼라 프로세서 시뮬레이터로 RUU(Register Update Unit)를 명령 윈도우로 사용한다[1]. RUU 는 비순차 이슈를 지원하기 위해서 디스패치된 명령을 저장하는 명령 윈도우의 기능과 명령 리카버리(Instruction Recovery)를 위해 프로그램의 순서를 유지하는 리오더 버퍼의 기능을 통합한 큐 구조의 장치이다. 전력 소모 측정을 위해 먼저 Simplescalar 의 동작 부분을 다음과 같이 기능별로 세분화 하였다.

- 명령 캐시, 데이터 캐시, L2 캐시, 명령 버퍼, TLB
- 명령 디코더, 디스패치 유닛(dispatch unit)
- RUU: 명령 윈도우 + 리오더 버퍼
- 레지스터 파일, 기능장치, 버스 I/O 버퍼
- 분기 예측기,BTB(Branch Target Buffer),값 예측기

세분화된 각 기능블록에 대해서 각 사이클 당 사용 빈도를 측정하는 동작 카운트 코드를 삽입하였고 이를 이용하여 프로세서의 동작 시 사이클 당 전력소모(활성 전력소모)를 측정하였다. 본 논문에서 사용한 Cai-Lim 모델[2]은 각 기능블록을 회로 종류에 따라 동적회로, 정적회로, 메모리, PLA, 클럭회로 영역으로 나누고 각 회로에 대한 단위면적 당 전력인 전력밀도(Power Density)와 각 기능블록에서 회로가 차지하는 면적을 계산하여 기능블록 별 전력을 계산하였고, 이들 기능블록 별 전력을 합산하여 사이클 당 전력소모를 측정하였다. 사이클 당 소모되는 기능블록 별 전력은 다음 그림 2 과 같이 계산하였다.

$BAP_i = AC_i * \sum_i (APD_j * AR_i)$
$BIP_i = \sum_j (IPD_j * AR_j)$

- BAP_i 사이클 i 에서 기능 블록의 활성전력(Block Active Power)
- BIP_i 사이클 i 에서 기능블록의 비활성전력(Block Inactive Power)
- AC_i 사이클 i 에서 기능블록의 동작 카운트(Active Count)
- APD_j 회로 j 에 대한 활성 전력 밀도(Active Power Density)
- IPD_j 회로 j 에 대한 비활성 전력밀도(Inactive Power Density)
- AR_j 회로 j 가 기능블록에서 차지하는 면적(Area)

그림 2. 기능블록 별 전력 소모 계산 방법

4. 실험 및 성능 분석

개발된 전력소모 측정 도구에 SPECint95 와 SPECint2000 벤치마크를 적용하여 전력 소모를 분석하였다. 시뮬레이션된 머신은 4,8,16 이슈 프로세서 구성에 각각 32KB 의 크기를 갖는 명령 및 데이터 캐시와 분기예측기로는 8K 엔트리를 갖는 혼합형 예측기를 사용하였다(gshare 와 bimodal 의 혼합형 예측기). 값 예측기로는 8K 엔트리 크기를 갖는 스트라이드와 2-단계 예측기를 통합한 혼합형 예측기를 사용하였다.

표 1. 벤치마크프로그램

Program	Input	Exce.Instr (milhon)	IPC	Brancl accurac	Value accuracy
compress	1000e2231	61	2.70	0.92	0.73
Gcc	jump.i	69	1.40	0.89	0.74
Go	5 9	200	1.40	0.84	0.67
ijpeg	tinyrose.ppm	97	3.15	0.95	0.75
Li	queen6.lsp	91	2.17	0.95	0.69
m88ksim	dhry.big.100	152	3.09	0.96	0.85
perl	scrabkl.in	66	1.88	0.96	0.70
vortex	persons.250	71	1.94	0.98	0.85
Mcf	smred.in,place	216	2.51	0.96	0.81
parser	smred.in	467	2.17	0.94	0.70
Vpr	smred.in	31	1.88	0.87	0.66
Average		126	2.20	0.92	0.74

본 연구는 한국과학재단 목적기초연구(2000-1-30300-008-3) 지원으로 수행되었음.

표 1 은 각 벤치마크 프로그램에 대해 입력값, 실행된 명령 수, IPC(Instrucrions Per Clock Cycle), 분기예측 정확도 및 값 예측 정확도를 나타낸 표로써 8-이슈인 경우를 기준으로 측정된 값이다.

그림 2 는 4,8,16 의 각 이슈에 대해 제안된 순차적인 복구 기법을 적용하여 값 예측을 한 경우에 값 예측을 하지 않은 경우에 비해 향상된 성능(speedup)을 보여주는 그림으로 마지막 평균값은 기하평균(geometric mean) 값이다. 그림에 나타난 바와 같이 4,8,16 이슈에 대해 각각 평균 1.1%, 5.1%, 5.4%의 성능 향상이 있음을 알 수 있다. 즉 이슈되는 명령 수가 커지면 값 예측을 이용한 모험적 실행이 더 효과가 이 있음을 알 수 있다.

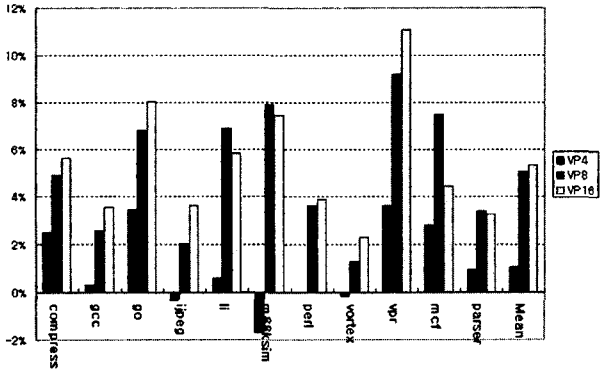

그림 2. 값 예측 시 성능 향상

그림 3 는 값 예측을 시도한 경우 값 예측을 하지 않는 경우에 비해 증가되는 각 사이클에서 소모되는 평균 전력을 측정한 결과이다. 4,8,16 이슈에 대해 각각 평균 9.9%,13.2%,15.8%로 전력소모가 증가됨을 알 수 있다. 값 예측으로 인한 전력소모의 증가는 두 가지 요인으로 분석될 수 있다. 값 예측기의 참조와 갱신 시의 전력소모 증가와 값 예측으로 인해 재 이슈된 명령들로 인한 전력소모 증가이다.

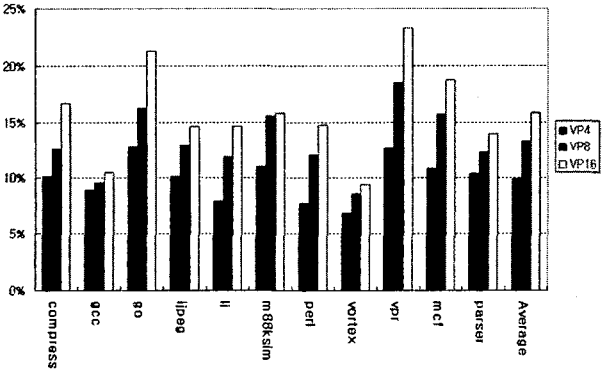

그림 3 값 예측 시 사이클 당 평균 전력소모 증가

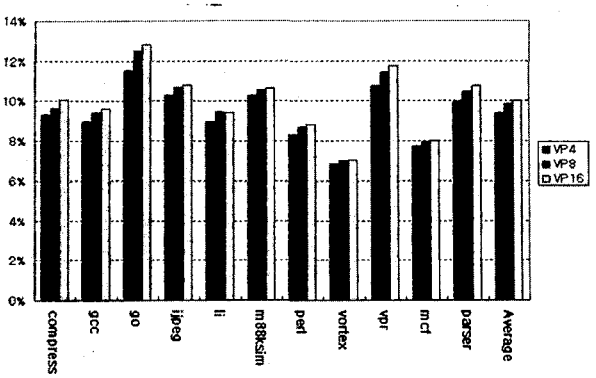

그림 4. 값 예측기의 참조와 갱신으로 인한 전력소모 비율

그림 4 는 전체 전력 소모에서 값 예측기의 참조와 갱신으로 인한 전력소모가 차지하는 비율을 보여 주는 그림으로 4,8,16 이슈에 대해 각각 평균 9.4%,9.8%,10.0%의 전력이 값 예측기로 인해 소모되었다. 즉 4 이슈인 경우 값 예측으로 인한 대부분의 증가된 전력소모가 값 예측기의 참조 및 갱신으로 유발된 것임에 비해 이슈 수가 8,16 으로 증가함에 따라 재 이슈로 부가된 전력소모가 증가되었다. 이는 이슈 수가 클수록 적극적인 모험적인 이슈로 예측 실패 시 부가되는 전력소모가 커짐을 의미한다.

5. 결론

본 논문에서는 값 예측기와 Cai-Lim 의 전력모델을 슈퍼스칼라 프로세서 사이클 수준 시뮬레이터인 SimpleScalar 3.0 툴셋에 삽입하여 전력 소모량을 측정하고 분석하였다. 분석 결과 값 예측 시 4,8,16 이슈에 대해 각각 평균 9.9%,13.2%,15.8%로 부가적인 전력소모가 측정되었다.

참고문헌

[1] D.Burger and T.Austin, The SimpleScalar tool set, version 2.0 ,Technical Report CS-TR-97-1342, University of Wisconsin, Madison, June 1997

[2] G.Cai and C.Lim, "Architectural level power/performance optimization and dynamic power estimation," Cool Chips Tutorial collocated with MICRO-32, Nov. 1999.

[3] M.Lipasti, and J.Shen, "Exceeding the limit via value prediction", Proceedings of the 29th International Symposium on Microarchitecture (MICRO-29), Dec. 1996.

[4] K.Wang, M.Franklin, "Highly accurate data value predictions using hybrid predictor," Proceedings of the 30th International Symposium on Microarchitecture (MICRO-30) , Dec. 1997.

[5] R.Bhargava and L.John, "Latency and energy aware Value prediction for high-frequency processors," In Proceddings of 16th ACM Interenational Conference on Supercomputing, pp. 45-56, June, 2002.

본 연구는 한국과학재단 목적기초연구(2000-1-30300-008-3) 지원으로 수행되었음.