수퍼스칼라 디지털 신호처리 프로세서에 대한 통계적 모의실험

이종복
한성대학교 정보통신공학과

Statistical Simulation for Superscalar DSP Processors

Jong-Bok Lee
Dept. of Information and Communications
Hansung University

Abstract

In this paper, statistical simulation is applied to a superscalar digital signal processor architecture using DSP kernel and DSP application benchmarks. As a result, the performance of a digital signal processor with several microarchitecture configurations can be estimated with the relative error of 3.7 \% on the average.

I. 서론

범용 마이크로 프로세서의 설계 과정에서 그 성능을 측정하기 위하여 트레이스 구동 모의실험이 주로 이용되고 있다. 그러나 트레이스 구동 모의실험은 그 높은 정확도에 불구하고 과다한 시간과 기억 공간이 필요하다. 최근에 이르러, 디지털 신호처리 프로세서에서도 수퍼스칼라나 VLIW 형태의 고성능 마이크로 아키텍처를 채택하기에 이르렀다.

본 논문에서, 신호처리용 수퍼스칼라 프로세서에 통계적 모의실험 모델을 적용하였다 [1,2]. 이를 위하여 DSP 벤치마크 프로그램에 대한 통계적 특성을 비롯하여, 명령어 캐쉬, 데이터 캐쉬 및 분기 예측 정확도에 대한 특성을 수집하여 이것을 바탕으로 새로운 명령어 트레이스를 합성하였다. 이것을 바탕으로 하는 통계적 모의실험은 통계적 프로화일을 기반으로 하기 때문에 보다 적은 수의 명령어를 가지고도 모의실험 속도가 빠르면서 동시에 높은 정확도를 나타낸다. 본 논문에서는 4 개의 DSP 커널과 4 개의 DSP 응용 프로그램을 대상으로 하여 그 성능을 기존의 트레이스 구동 모의실험으로 측정하였다. 또한, 통계적 프로화일링 기법을 이용하여 새로운 명령어 트레이스를 합성하고 통계적 모의실험을 수행하여 두 결과를 비교함으로써 그 정확도를 측정하였다.

II. 통계적 모의실험

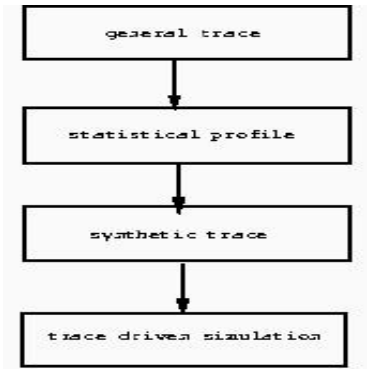

그림 1 통계적 모의실험의 전 단계

통계적 모의실험은 그림 1과 같이 크게 4 단계로 이루어진다. 단계 1에서는 기존의 일반적인 방법으로 각 벤치마크의 명령어 트레이스를 구한다. 단계 2에서는 통계적 프로화일링 기법을 이용하여 각 벤치마크 프로그램의 고유한 특성과 마이크로 프로세서의 고유한 특

성을 수집한다. 각 벤치마크 프로그램의 고유한 특성은 명령어의 유형별 분포, 각 유형별 명령어의 피연산자 개수의 분포 및 각 유형별 명령어의 데이터 종속 분포로 구성된다. 각 유형별 명령어의 데이터 종속은 각 명령어의 피연산자와 그 피연산자에 결과를 공급하는 명령어 간의 거리로 측정된다. 이러한 특성은 명령어 집합과 컴파일러에 의해서만 영향을 받는다. 한편, 마이크로 프로세서의 고유한 특성인 명령어 및 데이터 캐쉬 히트율, 분기 예측 정확도 및 분기 어드레스 캐쉬 히트율에 대한 정보도 수집된다. 통계적 프로화일링 단계는 반복될 필요가 없으며 일단 한번 얻어진 후에는 다양한 하드웨어에 대한 통계적 모의실험을 수행할 수가 있다.

단계 3에서는 0과 1사이의 난수를 발생시키고 이것을 통계적 프로화일링을 기반으로 하는 누적 분포 함수의 특정한 지점에 대응시킨다. 이 단계에서 명령어 유형과 피연산자의 개수, 명령어 유형 간의 레지스터 종속을 결정한다. 이 때, 분기나 스토어 명령어는 결과를 계산하지 않으므로 제외시켜야한다. 마이크로 아키텍쳐에 고유한 특성은, 각 명령어에 대하여 명령어 캐쉬의 히트 및 미스 여부 및 각 로드 명령어에 대하여 데이터 캐쉬의 히트 및 미스 여부, 그리고 각 분기 명령어에 대하여 분기 예측의 옳고 그름 여부를 결정한다. 이렇게 하여 새로운 명령어 트레이스를 합성한 후에, 디지털 신호처리 수퍼스칼라 프로세서를 통하여 통계적 모의실험을 수행하여 성능을 평가한다. 합성된 명령어 트레이스는 통계적 프로화일링을 기반으로 하므로 보다 적은 수의 명령어로 모의실험이 빠르게 수렴하여 그 결과를 신속히 얻을 수가 있다.

III. 수퍼스칼라 디지털 신호 처리 프로세서

본 논문에서는 범용 마이크로 프로세서 모의실험기인 SimpleScalar와 유사한 형태의 디지털 신호 처리 프로세서 모의실험기를 개발하였다 [3]. 통계적 수퍼스칼라 디지털 신호처리 모의실험기는 아키텍쳐를 구체적으로 수행할 필요가 없으며, 명령어 집합도 정수형, 로드, 스토어, 분기, 실수형 덧셈/뺄셈, 실수형 곱셈, 단일 정도 실수형 나눗셈 및 2 배 정도 실수형 나눗셈의 8 개 명령어로 구성된다. 명령어 간의 데이터 종속은 모두 확률적으로 발생시키기 때문에 레지스터 파일도 존재하지 않는다. 그러나 매 싸이클마다 모든 명령어는 명령어 윈도우, 연산유닛, 리오더 버퍼를 차례로 거친다. 구체적인 프로세서의 사양은 표 1에 나타낸 것과 같다. 명령어는 비순차 수행될 수 있지만, 순차적으로 완료된다. 분기 예측을 위하여 이중 적응형 분기 예측법을 채택하였으며, 이 때 14 비트의 분기 히스토리 레지스터와 16 K 패턴 히스토리 테이블을 이용하였다 [4].

정수형, 로드, 스토어, 분기 및 실수형 덧셈/뺄셈의 지연은 모두 1 싸이클이며, 실수형 곱셈의 지연은 3 싸이클이다. 또한 단일 및 2 배정도 실수형 나눗셈의 지연은 각각 6 싸이클과 9 싸이클이다. 1차 캐쉬와는 달리, 2 차 캐쉬는 모의실험에 포함되지 않았다.

Ⅳ. 모의실험 환경

표 2에 나타낸 것과 같이, **UTDSP** (University of

아키텍쳐 사양		값		
윈도우의 크기		16	32	64
인출율		4	8	16
이슈율		4	8	16
퇴거율		4	8	16
연산 유닛	정수형ALU	4	8	16
	로드/스토어	2	4	8
	실수형 덧셈	1	2	4
	실수형 나눗셈	1	2	4
1차 명령어 캐쉬		32KB, 2차 세트연관 32B 블록 10 싸이클 미스 지연		
1차 데이터 캐쉬		32KB, 직접매핑 32B 블록 10 싸이클 미스 지연		
분기 예측기		16K엔트리,6 싸이클미스지연		

표 1 아키텍쳐 사양

벤치마크	설명
FFT	1024 점의 복소수 FFT
FIR	64 점의 256 탭 FIR 필터
IIR	64점의 4-직렬 IIR biquad 필터
LMSFIR	64 점의 32 탭 LMS 적응 필터
COMPRESS	DCT를 이용한 128x128 영상 이미지의 압축
EDGE DETECT	256 그레이 레벨의 128x128 영상 이미지의 에지 추출
JPEG	16 비트 데이터의 32 비트 데이터로의 확장
LPC	선형예측코딩 인코더

표 2 UTDSP 벤치마크

Toronto Digital Signal Processing) 벤치마크에서 4개의 DSP 커널과 4 개의 DSP 응용 프로그램을 선정하여 모의실험의 입력 프로그램으로 이용하였다. 이 프로그램들은 운영체제 SunOS 5.6 하의 Sun UltraSPARC에서 C 컴파일러를 이용하여 컴파일하여 실행화일을 얻었으며, **Shade**를 이용하여 5천만개의 명령어 트레이스를 발생시켰다 [5]. 통계적 트레이스를 대상으로 통계적 모의실험을 수행할 때는 성능의 표준편차가 1 % 이내를 만족할 때 수렴된 것으로 간주하였다.

V. 모의실험 결과

벤치마크	기본블럭	정수형	실수형	로드	스토어	분기어	분기예측(%)	명령어 캐쉬(%)	데이터 캐쉬(%)
FFT	6.4	49.4	1.9	25.1	8.3	15.4	95.0	94.6	98.1
FIR	5.8	38.4	0.5	26.9	16.9	17.3	89.5	92.7	96.7
IIR	4.6	58.2	0.3	13.6	6.2	21.7	91.4	93.3	94.6
LMS FIR	5.6	42.0	2.6	26.4	11.6	17.4	90.3	95.5	95.7
comp	6.0	38.6	0.1	27.2	17.5	16.7	91.5	97.9	98.7
edge detect	6.0	38.4	0.0	27.3	17.7	16.7	91.4	96.5	98.7
JPEG	4.8	46.7	0.1	25.1	7.4	20.7	96.5	99.9	87.2
LPC	6.2	38.0	5.1	29.5	11.9	15.6	98.7	81.1	95.5

표 3 UTDSP 벤치마크의 통계적 기본 특성

통계적 프로화일링에 의한 결과가 표 3에 요약되어있다. 기본 블록의 크기는 5.6이며, 대부분의 벤치마크에 걸쳐서 분기 예측 정확도는 89 %에서 95 %의 범위를 나타내었다. 1차 명령어 캐쉬 히트율은 93 %에서 99 %의 범위를 나타내지만, 예외적으로 **lpc**가 극히 저조한 히트율을 기록하였다. 1차 데이터 캐쉬도 jpeg를 제외하고 94 %에서 98 %의 높은 히트율을 보였다. 그림 2에 **fft** 벤치마크에 대하여 정수형 명령어간 종속 거리의 분포율을 나타내었다. 정수형 명형어의 피연산자 레지스터 2 개에 대하여 각각 2 개의 히스토그램이 필요하다. 명령어간 종속거리가 증가할 수록, 그 분포율이 급격하게 감소하는 것을 알 수 있으며, 통계적 모의실험의 정확도에 있어서 명령어의 병렬성을 나타내는 가장 중요한 요소이다.

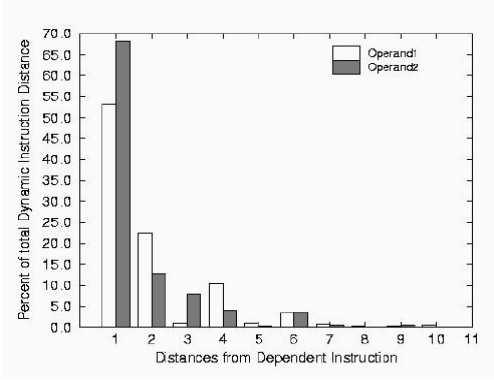

그림 2 **fft**에서 정수형 명령어의 종속거리

본 논문에서 프로그램의 성능을 측정하기 위한 지표로 싸이클 당 명령어의 실행률을 나타내는 **IPC** (Instructions Per Cycle)을 이용하였다. IPC의 상대오차는 수식 1과 같이 표시할 수 있다. 각 벤치마크에 대하여 상대오차를 구한 후 평균을 취하여 전체 벤치마크에 대한 정확도를 가늠하는데 사용하였다.

$$\Delta_{IPC} = \frac{|IPC_{synth} - IPC_{real}|}{IPC_{real}}$$

수식 1 IPC 상대오차

그림 3부터 그림 5에서 각 DSP 벤치마크에 대하여 일반적인 트레이스 구동 모의실험에 의하여 측정한 성능과 통계적 모의실험에 의하여 측정한 성능을 비교하여 나타내었다. 각 벤치마크 프로그램의 흰색 막대 그래프는 일반적인 트레이스 구동 모의실험에 의하여 측정한 성능이다.

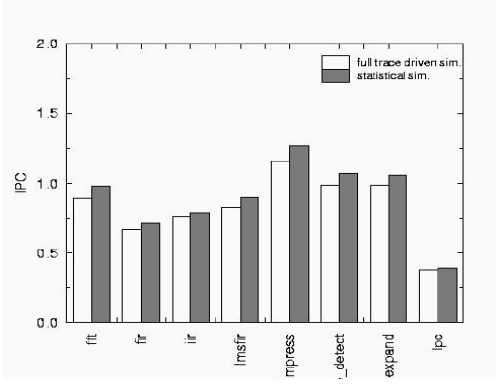

그림 3 윈도우 크기가 16일 때의 성능 비교

그림 3에서 **lpc** 벤치마크는 실수형 명령어의 비율이 높고 명령어 캐쉬 히트율이 81 %로 매우 낮기 때문에

0.42 IPC에 불과한 성능을 기록하였다. 이 때 모든 프로그램의 성능에 대한 IPC의 조화평균은 0.80에서 1.04를 나타내었다. . 각 벤치마크의 검은색 막대 그래프는 통계적 모의실험에 의하여 측정한 성능을 나타낸다. 윈도우의 크기가 16이고 명령어 인출율이 4 일 때 상대오차의 평균은 7.0 %를 기록하였다.

윈도우의 크기가 32로 증가하고 명령어 인출율이 8일 때 상대오차의 평균은 3.1 %로 감소하였으며, 윈도우의 크기가 64로 증가하고 명령어 인출율이 16일 때 상대오차의 평균은 1.2 %로 감소하였다. 윈도우의 크기가 16에서 64로 4 배로 증가하였을 때, 조화평균은 1.3

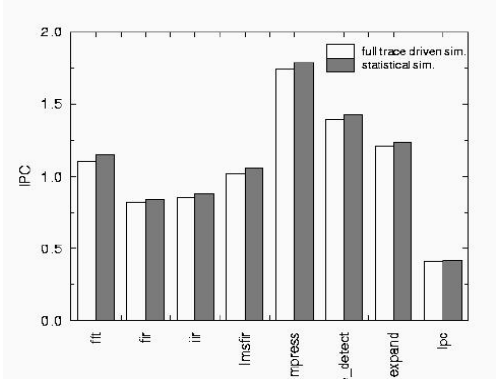

그림 4 윈도우의 크기가 32일 때 성능의 비교

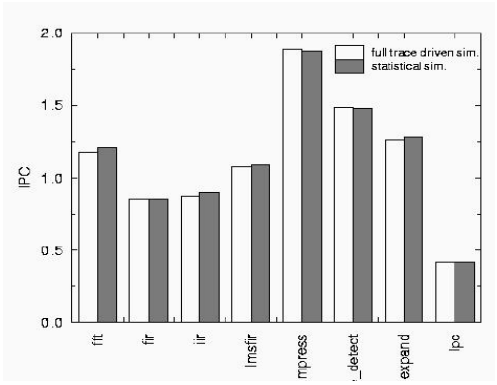

그림 5 윈도우 크기가 64일 때의 성능의 비교
배 향상되었다

상대오차의 평균은 윈도우의 크기와 명령어 인출율이 증가할 수록 감소하였는데, 그 이유는 작은 명령어 윈도우에서는 상대적으로 작은 종속거리를 갖는 명령어들로 구성되어 성능이 과다 측정되었기 때문이다. 본 논문에서 시행한 3 가지 서로 다른 아키텍쳐 사양에 대하여 통계적 모의실험을 시행한 결과 상대오차의 평균으로 3.7 %를 얻었다.

VII. 결론

본 논문에서는 DSP 커널과 DSP 응용 프로그램을 이용하여 디지털 신호 처리 수퍼스칼라 프로세에 대하여 통계적 모의실험을 수행하였다. 통계적 모의실험에 의한 성능을 기존의 일반적인 트레이스 구동 모의실험의 성능과 비교한 결과. 3.7 %의 상대오차를 얻을 수 있었다. 따라서, 범용 마이크로 프로세서 뿐만이 아니라 수퍼스칼라 디지털 신호처리 프로세서에 대하여통계적 모의실험이 유용함을 확인할 수 있었다..

통계적 모의실험의 정확도를 더욱 높이기 위한 방안을 살펴보면 다음과 같다. 통계적 프로화일링을 각 벤치마크 프로그램 전체가 아닌 각 벤치마크 프로그램의 기본블럭 단위 또는 서로 다른 프로그램의 단계마다 시행하는 방법이다. 마지막으로, 통계적 프로화일링 과정에 프로그램의 흐름도에 관련된 정보를 포함시켜서, 프로그램의 특성을 더욱 자세히 반영할 수 있다.

참고문헌

[1] R. Carl and J. E. Smith, "Modeling Superscalar Processors via Statistical Simulation," Workshop on Performance Analysis and Its Impact on Design, Oct. 1996.

[2] S. Nussbaum and J. E. Smithm "Modeling Superscalar Processors via Statistical Simulation." International Conference on Parallel Architectures and Compilation Techniques," Sep. 2001, pp.15-24

[3] T. Austin, E. Larson, and D. Ernst, "Simplescalar : An Infrastructure for Computer System Modeling," Computer, Feb. 2002, vol. 35, pp. 59-67.

[4] T-Y. Yeh and Y.N. Patt, "Two-Level Adaptive Branch Prediction." The 24th SCM/IEEE International Symposium and Workshop on Microarchitecture, May. 1992, pp.124-134.

[5] Introduction to Shade Sun Microsystems. Inc., Jun. 1997.

[6] The SPARC Architecture Manual Version 9, SPARC International Inc., Jul. 2003.