

고성능 내장형 마이크로프로세서의 분기 예측기 구현 및 성능 대비 비용 분석

신상훈^o 최 린
고려대학교 전자컴퓨터공학과
{kaisshin^o, lchoi}@korea.ac.kr

Implementation of a Branch Predictor and Its Cost Per Performance Analysis for a High Performance Embedded Microprocessor

SangHoon Shin,^o Lynn Choi
Dept. of Electronics and Computer Engineering, Korea University

요 약

EISC ISA를 기반으로 한 64 비트 고성능 내장형 마이크로프로세서 AE64000의 효과적인 성능 향상을 위해서 비용 대비 성능 향상이 우수한 분기 예측 기법을 도입하여 AE64000 파이프라인에 적합한 분기 예측기를 추가로 설계하고 SPECint 벤치마크 및 타 내장형 벤치마크의 성능 분석 시뮬레이션을 통해 최적의 분기 예측기의 구조를 결정하였다. AE64000에서 LERI 명령 처리를 위해 AE64000 파이프라인에 추가된 독특한 IFU에 의하여 복잡성을 갖지만, IF 단계의 PC 대신에 IFU 단계의 PrePC를 이용하여 분기 명령을 명령어 prefetch 단계에서 예측함으로써, 올바른 분기 예측시 분기로 인한 손실을 제거할 수 있다. 결과적으로 최종 선정된 최적의 분기 예측기는 Verilog로 구현하여 AE64000 프로세서 코어 모델과 통합 합성 하였고 아울러 추가되는 면적과 최종 목표 클럭에 동작하기 위한 타이밍 분석을 통해 최종 생산에 적합하도록 설계된 분기 예측기의 기능 및 타이밍 검증을 수행하였다. 최종 구현된 분기 예측기는 프로세서 칩 전체의 1% 미만의 비용으로 최고 12%의 성능 향상을 달성하여 성능 대비 면적의 효율성에서 높은 결과를 보였다.

I. 서 론

AE64000은 (주)에이디칩스에서 고성능 내장용 환경을 목표로 개발된 64비트 마이크로프로세서로서 5단계 파이프라인으로 구성되며, 명령어 캐쉬와 데이터 캐쉬가 분리된 하바드 아키텍처를 사용하며 64 비트 레지스터를 분할하여 8/16/32/64 비트의 MAC 연산을 가능하게 하여 SIMD 형태의 DSP와 FP 연산을 지원한다. 또한, EISC ISA를 구현하고 있어 LERI 명령을 효율적으로 처리하기 위해 IFU(Instruction Folding Unit)를 내장하는데 이로 인해 분기예측을 어렵게 한다.

본 논문은 AE64000에 비용 대비 성능이 가장 우수한 분기 예측기를 구현하기 위해 사이클 단위의 정확한 시뮬레이션이 가능한 명령어 수준의 AE64000 시뮬레이터를 개발하였고 SpecINT 벤치마크에서의 Execution-driven 시뮬레이션을 통해 Last-Time Predictor, Bimodal Predictor, G-share Predictor 등 3 가지 분기 조건 예측 기법과 목적 주소 캐쉬(target address cache)의 구성을 변화시켜 성능을 분석하였다. 이러한 성능 분석을 통해 후보가 되는 분기 예측기를 Verilog로 구현하였고 프로세서 칩과 합성하여 추가되는 게이트 수를 분석 등, 여러 상황을 고려한 최적의 분기

예측기 구조를 결정하였다. 최종 선정된 분기 예측기는 캐쉬와 FP/DSP 장치를 포함한 AE64000 Mustang 프로세서 칩 전체의 0.6% 정도의 면적으로 (프로세서 코어의 6.24% 면적) 최고 12%의 성능 향상을 달성하여 성능 대비 면적의 효율성에서 높은 결과를 보였다.

II. AE64000

AE64000은 64비트의 프로세서이지만, 16비트로 고정된 길이의 명령어를 사용한다. 컴파일러에서는 operand를 분할하여 LERI라는 특유의 명령어와 함께 12비트씩 저장한다. 최고 64비트의 길이가 되는 데이터는 5개의 LERI 명령어와 4비트의 immediate 데이터를 사용하는 1 개의 일반 명령어로 처리된다. LERI 명령어 처리에 따른 성능저하를 막기 위해 IFU(Instruction Folding Unit)는 파이프라인의 초기에서 명령어 인출(fetch)과 동시에 LERI 명령어들을 처리(folding)하고 LERI가 제거된 명령어들을 파이프라인 코어로 보낸다. IFU에서 명령어를 4개씩 인출(fetch)하는데, LERI 제거를 원활하게 하기 위해 IFU내에는 12개의 명령어를 저장하는 버퍼를 가지고 있다. IFU는 LERI 명령어들을 제거하는데 2

clock의 시간이 필요로 한다. 분기 명령이 디코딩(ID) 단계에서 인식이 되면 이미 인출(IF) 단계에 있는 명령어를 포함하여 3 clock의 손실이 있게 된다.

파이프라인 코어에 LERI가 제거된 후 실제 실행하는 명령들의 주소를 갖고 있는 PC와는 별도로, IFU내에는 명령어 캐쉬로 부터 인출 주소를 갖고 있는 PPC(pre-PC)가 있다.

III. 제안된 AE64000 분기 예측 기법의 소개

3.1 2bit Bimodal Branch Predictor

하나, PPC에서 들어온 정보를 가지고 그에 따라서 예측과 함께 예측된 분기 목적 주소가 IFU 내에 캐쉬 구조의 TAC(Target Address Cache)에 저장된다.

그림 1은 AE64000 파이프라인 프로세서 내부에 구현한 2bit Bimodal 분기 예측기의 블록 다이어그램이다.

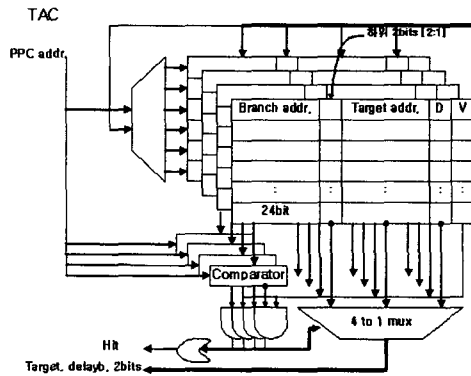


그림 1. Branch Predictor Diagram for AE64000

둘, 예측하고자 하는 분기 명령이 이전에 예측된 분기 목적주소 이전에 있는 것인지 확인을 위한 Comparator가 필요하다.

셋, 이전에 분기 명령어였음을 확인하기 위한 1K 개의 entry를 가진 분기예측 테이블(Branch Prediction Table)을 두고 IFU 주소의 하위비트를 받아서 분기 조건 예측 테이블을 검색하고 그 결과(Taken or Not-taken)를 출력한다. 보다 정확하게 분기임을 예측하기 위해서 분기조건 테이블(Branch Pattern Table: BPT)의 각 엔트리에 분기 명령의 주소를 태그로 저장해야 하지만 여기서 구현된 예측기는 분기 명령의 주소를 저장하는 태그를 구현하지 않고 분기 명령의 주소를 이용하여 해당 엔트리의 값을 변경하는 방법을 택하여 구현을 하였다.

3.2 1bit Last-Time Branch Predictor(LT)

기본적인 구조는 Bimodal Predictor와 같으며 Direct-Mapped Cache 구조로 구현하였다. 여기서는 BPT를 사용하지 않으며 LT 내에서의 entry의 크기에 변화를 주고 예측정보(Taken or Not-taken)을 표시하여 예측정보에 따라서 목적주소(Target Address-Buffer)를 접근토록 하여 entry에 변화를 주어 실행을 하였다.

3.3 G-share Branch Predictor

Taken or Not-taken의 1bit 정보를 가진 shift register인 Global Register(GR) 와 PPC의 하위 10bit를 서로 XOR한 결과를 가지고 BPT를 접근하여 분기를 예측하도록 한다.

IV. 분기 예측기의 성능 대비 비용 분석

4.1 비용비교 (Bimodal & Last-Time Predictor)

Bimodal 예측기와 LT 예측기의 환경에 대한 변화를 주었을 때 RTL 수준에서의 비교를 아래에서 보인다.

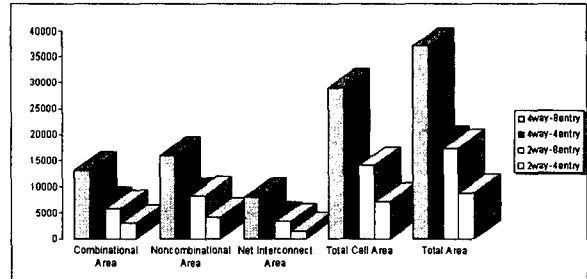


그림 2. way-entry 대한 변화에 대한 Bimodal Predictor의 gate수 (개)

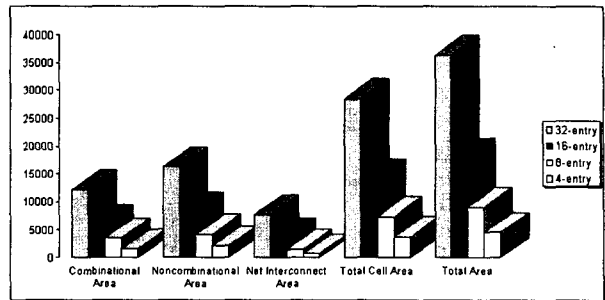


그림 3. Entry 변화에 대한 Last-Time Predictor의 gate수 (개)

AE64000 프로세서 칩 전체에서 차지하는 총 gate의 수는 132만 9천개이며 분기 예측기(LT-8entry : 8981 개)를 추가할 경우 칩을 차지하는 총 gate의 수는 133 만 8천여개가 되며 LT-8entry 분기 예측기가 추가될 경

우 0.682.7%의 면적이 추가된다.

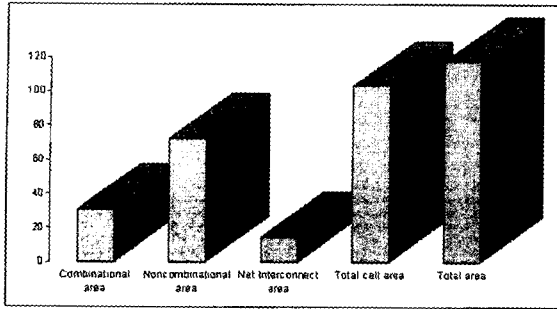


그림 4. G-share Predictor의 Gate Level 정보

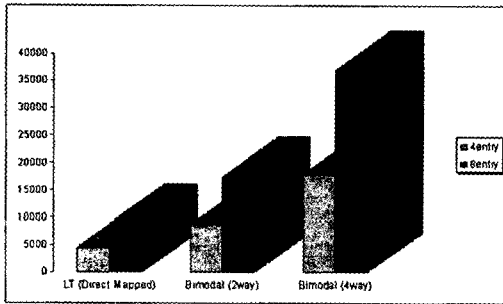


그림 5. entry당 LT와 Bimodal의 크기비교

칩 전체의 면적을 고려할 경우 1% 채 되지 않는 8entry TAC last-time 분기 예측기의 삽입이 가장 효율적이며 hit ratio는 95.6% 이상이며 성능은 12% 이상 향상시킬 수 있음을 확인하였고 내장형을 목적으로 개발된 프로세서인 점을 감안하여 설계된 분기 예측기는 칩 전체에서 차지하는 면적이 작아서(0.6%) 위에서 설명한 다른 예측기와 비교분석을 통해서 4way-set associative 8entry TAC 구조의 2bit Bimodal Branch Predictor를 최종 선택하여 적용하였다.

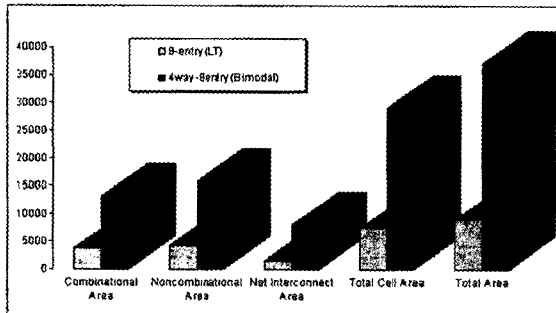


그림 4. 최종 선택된 분기예측기의 비교 (개)

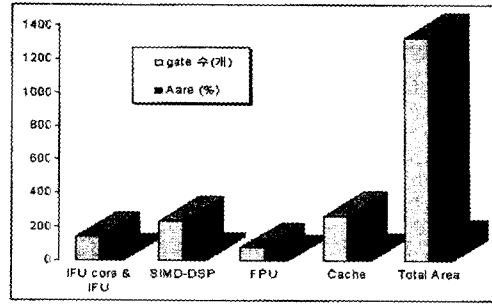


표 1. AE64000 프로세서의 주요 모듈의 비교

IV. 결론

AE64000에서 여러 가지 비용적 측면을 고려하여 가장 적합한 분기예측기를 설계하였고 C language를 이용한 시뮬레이터를 제작하여, 원활하게 동작함을 확인했다.

AE64000 파이프라인의 좀더 전체적인 성능의 개선을 위해서 슈퍼스칼라, 슈퍼파이프라이닝, 비순차적 실행 기법 등 여러 가지 마이크로기법들을 검토한 결과 파이프라인의 변화를 최소화할 수 있는 슈퍼파이프라이닝 기법을 도입한다면 현재의 프로세서보다 더 나은 성능을 발휘할 수 있다는 것을 여러 가지 자료들을 통해서 확인하였다.

본 연구는 한국반도체연구조합(COSAR) 및 반도체설계교육센터(IDEC)의 지원을 받아 수행되었습니다.

참고문헌

- [1] S. McFarling, "Combining Branch Predictors", DEC WRL Technical Note TN-36, June 1993
- [2] Brian K. Bray, M. J. Flynn, "Strategies for branch target buffers", Proceedings of the 24th annual international symposium on Microarchitecture September 1991
- [3] T. Y. Yeh, D. Marr, Y. Patt, "Increasing the Instruction Fetch Rate via Multiple Branch Prediction and a Branch Address Cache", in Proceeding of the 1993 International Conference on Supercomputing, pp. 67-76, 1993
- [4] S. H. Lee, I. K. Kim, L. Choi, "Branch Predictor Design and Performance Estimation for a High Performance Embedded Microprocessor", In the proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC 2003) Tokyo, Japan, 2003