

# 에너지-지향 시뮬레이터의 분석 및 설계

고광만, 김성진  
 상지대학교 컴퓨터정보공학부  
 e-mail:{kkman, sjkim}@sangji.ac.kr

## Analysis and Design of the Energy-oriented Simulator

Kwang-Man Ko, Sung-Jin Kim  
 School of Computer Information Engineering, Sang-Ji University

### 요 약

새로운 임베디드 프로세서를 설계하거나 재설계하는 경우에 실제 하드웨어로 구현하기 전에 프로세서의 다양한 성능확인 및 개선을 위해 소프트웨어적으로 검증할 수 있는 시뮬레이터와 같은 실험환경의 구축은 하드웨어 설계자 및 임베디드 소프트웨어 개발자에게 중요한 요소이다. 본 논문에서는 임베디드 프로세서 설계자의 아키텍처 탐색과 성능측정 환경을 지원하기 위해 임베디드 프로세서를 위한 시뮬레이터를 인스트럭션-셋 캄파일 방식으로 설계한 후 모델을 제시한다. 또한 시뮬레이터를 구성하는 각 요소에 대한 기능적, 연결성 기술을 통해 실제적인 개발에 활용될 수 있도록 한다.

### 1. 서론

임베디드 프로세서 설계자는 다양한 목적에 따라 기존 임베디드 프로세서를 확장하거나 새로운 프로세서 설계를 시도한다. 이러한 시도의 결과를 하드웨어로 구현하기 전에 프로세서의 다양한 성능 확인과 개선을 위해 소프트웨어적으로 검증할 수 있는 시뮬레이터와 같은 실험 환경의 구축은 하드웨어 설계자 및 임베디드 소프트웨어 개발자에게 중요한 요소이다[1]. 또한 최근에 임베디드 프로세서의 에너지 소비를 측정하고 추적할 수 있는 실험 환경을 구축은 많은 관련 연구와 더불어 수요가 지속적으로 제기되고 있는 분야이다.

본 논문에서 설계하는 에너지-지향 시뮬레이터(Energy-oriented Simulator)는 아키텍처에 대해 소프트웨어적으로 탐색하고 개선하기 위해 실행 속도, 메모리 사용 등과 같은 전통적인 성능 측정과 더불어 어플리케이션의 에너지-소비량, 에너지-소비 발생의 원인, 에너지-소비 추적 등의 결과를 생성한다. 또한 성능분석 결과에 대한 지속적인 피드백을 통해 양질의 코드 생성, 에너지 최적화, 아키텍처 탐색과 설계에 편리한 시뮬레이터를 개발한다.

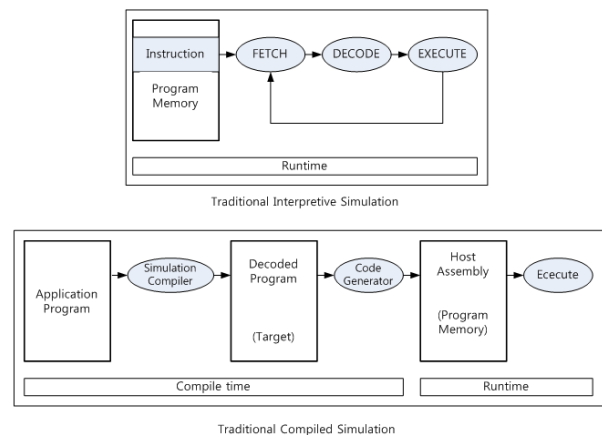
### 2. 관련연구

#### 2.1 재목적 시뮬레이터

ADL로부터 설계되는 아키텍처는 실제 하드웨어 개발 전에 소프트웨어적으로 아키텍처에 대한 탐색을 위해 시뮬레이터 및 디버거를 구축하므로써 많은 개발 비용과 시간을 줄일 수 있다. 특히, 최적의 성능을 갖춘 하드웨어 설계 및 짧은 하드웨어 라이프-사이클에 대처하기 위해서

는 ADL로부터 시뮬레이터, 디버거 등과 같은 소프트웨어적인 실험 환경을 구축하는 것이 효과적이다. 시뮬레이터의 구축은 아키텍처에 대한 추상화 수준에 따라 인스트럭션-셋 시뮬레이션 모델과 사이클-정확 시뮬레이터 모델로 구분된다. 다양한 프로세서에 대한 시뮬레이션이 가능한 재목적 시뮬레이터는 임베디드 SoC 분야에서 하드웨어, 소프트웨어 통합 설계 분야에서 중요하게 사용되고 있다.

시뮬레이터 구성 모델은 추상화 수준에 따라 인스트럭션의 모델링을 통해 기능적인 시뮬레이션이 핵심이 되는 인스트럭션-셋 시뮬레이터(ISS)와 보다 자세한 타이밍 정보가 요구되는 사이클-정확성(cycle-accurate) 시뮬레이터로 구분된다[2].



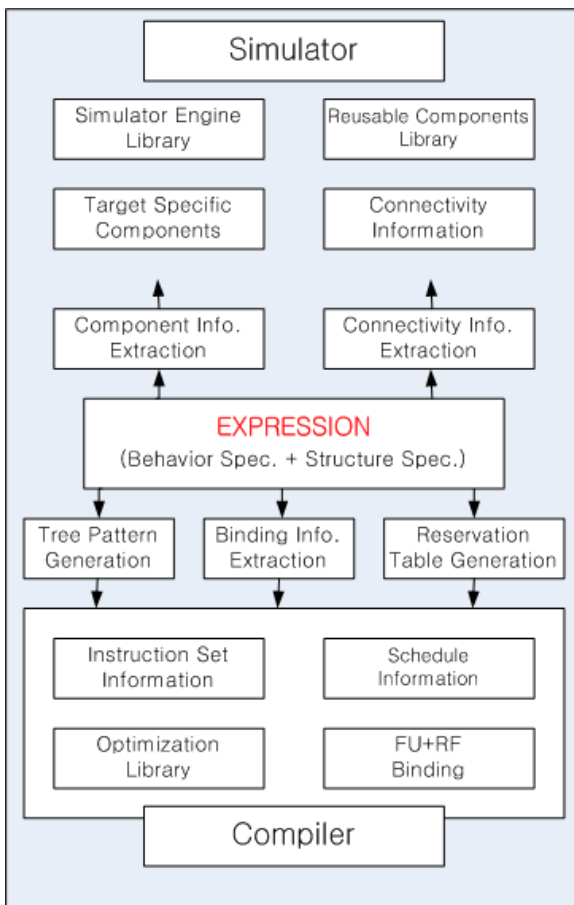
[그림 1] 시뮬레이터 모델 비교

인스트럭션-셋 시뮬레이터에서는 인터프리티브 시뮬레

이션(interpretive simulation), 컴파일 방식 시뮬레이션(compiled simulation), 혼용 시뮬레이션(mixed simulation) 모델로 세분화되며 [그림 1]과 같이 인터프리티브 방식에 갖는 융통성을 보존하면서 시뮬레이션 속도를 개선하기 위해 컴파일 방식 시뮬레이션 관련된 국내외적인 연구 시도가 [3]을 시작으로 Facile[4], JIT-CCS[5], IS-CS[6,7], FacSim[8] 등에서 선행되었다.

2.2 EXPRESSION의 시뮬레이터

EXPRESSION은 구조/행위 정보를 가진 LISP-like한 아키텍처 기술 언어로부터 컴파일러(EXPRESS)와 시뮬레이터(SIMPRESS)를 [그림 2]와 같이 생성하며 MDes[9]를 기반으로 계층적 메모리 구조 표현과 프로세서의 구조를 시각적으로 확인할 수 있는 기능을 보완하였으며 MIPS R4000 프로세서에 대한 기술 결과를 제시하였다. 메모리 계층과 파이프라인 메커니즘, 데이터 전송경로에 대한 아키텍처 정보를 기술할 수 있지만 불규칙한 하드웨어의 특성을 모델링하는데 제한적인 단점을 갖는다. 또한 MIPS R4000에 대한 EXPRESSION 기술을 통해 생성된 컴파일 결과와 시뮬레이터의 결과가 미흡하며 다른 프로세서를 기술하는데 많은 제약점이 지적되고 있다[10].

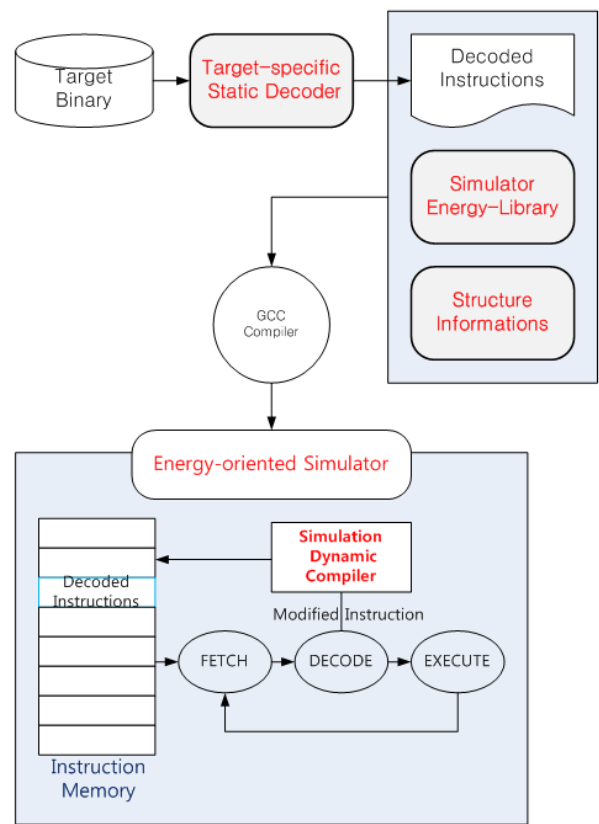


[그림 2] SIMPRESS 시뮬레이터 구성 요소

3. 에너지-지향 시뮬레이터 모델 설계

3.1 전체 시스템 구조

에너지-지향 시뮬레이터는 프로세서에 대해 소프트웨어적으로 탐색하고 개선하기 위해 디버거 기능을 갖춘 전통적인 성능 측정과 더불어 에너지 소비 측정, 예측, 추적 동작을 수행하며 [그림 3]과 같은 구조를 가지고 있다. 이 가능한 에너지-지향 시뮬레이터를 생성한다. 추출된 정보를 이용하여 생성되는 에너지-지향 시뮬레이터는 디코드된 인스트럭션이 인스트럭션 메모리에 탑재된 후 페치, 디코드, 실행 과정을 반복하게 되며 페치된 인스트럭션에 대한 변경이 감지되는 경우에는 재컴파일 과정을 통해 인스트럭션 메모리에 탑재된 후 실행된다.



[그림 3] 에너지-지향 인스트럭션-셋 시뮬레이터 구조

3.2 구성요소

디코더(decoder)는 타겟 프로세서의 바이너리 파일을 시뮬레이터에서 실행할 수 있는 인스트럭션 스트림으로 변환하는 동작을 수행한다. 따라서 빠른 시간에 디코딩을 수행하는 것은 전체적인 시뮬레이션 속도를 신속하게 수행하는 핵심요소이다. 또한 빠른 디코딩을 수행하기 위해서는 시뮬레이터 인스트럭션의 형식이 시뮬레이션 속도를 결정하는 중요한 요소이다. 본 논문에서는 다양한 바이너리 파일에 대한 빠른 디코딩이 가능하도록 포괄적 인스트

러션 포맷을 설계하여 시뮬레이션의 속도 향상에 기여한다.

시뮬레이터 에너지-라이브러리는 프로세서에 독립적인 시뮬레이터의 구성과 동작을 위한 정보를 가지고 있다. 또한 구성되는 시뮬레이터가 인스트럭션 스트림의 실행에 대해 에너지 소비량 측정, 에너지 소비 요소 추적, 에너지 소비 요소 예측 등에 활용되는 다양한 정보를 가지고 있다. 구조 정보를 시뮬레이터를 구성하는 정보로서 파이프라인과 어플리케이션의 데이터 흐름에 관한 정보를 저장하고 있으며 시뮬레이터 구성에서 중요한 역할을 수행한다.

디코딩된 프로그램 코드, 시뮬레이터-에너지 라이브러리, 구조 정보는 모듈화된 C++ 형식으로 출력되며 GNU 컴파일러에 의해 실제적인 시뮬레이터로 변환된다. 시뮬레이터는 시뮬레이터의 인스트럭션 메모리로부터 인스트럭션을 읽어들이 페치, 디코드, 실행의 과정을 반복하게 된다. 디코드시에 인스트럭션에 변경이 감지되면 시뮬레이터 컴파일러를 통해 재컴파일을 수행한 후 인스트럭션 메모리에 저장한 후 페치, 디코딩, 실행의 과정을 반복한다.

#### 4. 결론 및 향후연구

임베디드 프로세서의 새로운 등장, 변경 등은 실제적으로 하드웨어 설계전에 소프트웨어적으로 탐색하고 성능을 측정할 수 있는 시뮬레이터의 지원이 필수적이다. 본 논문에서는 기존 인스트럭션-셋 컴파일 방식의 시뮬레이터를 고찰한 후 시뮬레이터의 속도 개선과 에너지 소비, 추적, 예측이 가능한 시뮬레이션을 설계하였다. 이를 위해 바이너리에 대한 디코딩을 원활히 수행할 수 있는 포괄적 인스트럭션 포맷을 설계한 후 실제 디코딩을 위한 정적 디코딩 알고리즘을 구현하였다. 향후 보다 구체적인 실험결과를 제시할 예정이며 빠른 시뮬레이터 엔진 개발을 위한 연구를 진행할 예정이다.

#### 참고문헌

- [1] J. Rehg, "A Compilation Framework for Power and Energy Management on Mobile Computers", Int. Workshop on Language and Compilers for Parallel Computing, August, 2001.
- [2] Prabhat Mishra, Nikil Dutt, Processor Description Languages, Morgan kaufmann, 2008.
- [3] J. Zhu et. al, "A Retargetable Ultra-fast Instruction Set Simulator", DATE, 1999.
- [4] E. Schnarr et. al., "Facile: A Language and Compiler for high-performance Processor Simulators", ACM PLDI, 2001.
- [5] A. Nohl, G. Braun, et. al., "A Universal Technique for Fast and Flexible Instruction-set Architecture Simulation", In Proc. of DAC, 2002.
- [6] M. Reshadi, et. al., "Instruction Set Compiled Simulation: A Technique for Fast and Flexible Instruction-set Simulation", In Proc. of DAC, 2003.
- [7] M. Reshadi, et. al., "An Efficient Retargetable Framework for Instruction-set Simulation", In Proc. of CODES+ISSS, 2003.
- [8] Jaejin Lee, et. al., "FaCSim: A Fast and Cycle-Accurate Architecture Simulator for embedded Ststems", ACM LCTES'08, 2008.
- [9] J. C. Gyllenhaal et. al., The Mdes User Manual, Technical Report, Trimaran Release.: <http://www.trimaran.org>. 1999.
- [10] P. Mishra, A. Shrivastava, N. Dutt, "Architecture Description Language(ADL)-driven Software Toolkit Generation for Architectural Exploration of Programmable SOCs", ACM Trans. on Design Automation of Electronics Systems, Vol. 11, No. 3, 2006.