동적인 프로세서 모델 선택에 의한 효율적인 코시뮬레이션 방법

고현우, 배종열, 정정화
서울시 성동구 행당동 17 한양대학교 CAD 및 통신 회로 연구실
Tel : (02) 2290 - 0562, e-mail : choice@hymail.hanyang.ac.kr

Efficient Co-simulation Method with Dynamic Selection of Processor Model

Hyun-Woo Koh, Jong-Yeol Bae, Jong-Wha Chong
CAD & C.C. Lab., Dept. of Electronic Eng.,
Hanyang University., 17 HaengDang-Dong, SeongDong-Gu, Seoul, Korea
Tel : (02) 2290 - 0562, e-mail : choice@hymail.hanyang.ac.kr

Abstract

In this paper, the efficient HW/SW co-simulation method which selects the ISA model dynamically is proposed. Because the ISA models with only fixed accuracy have been used in previous co-simulation environment, it may result in bad performance in speed or accuracy. In the proposed method, the cycle accurate ISA model is used in the case that the states of the detailed system are to be inspected. In other case, instruction-based model is executed in order to accelerate the simulation speed. The proposed dynamic model selection can be done by setting the conversion point in the application code before the simulation starts. The experiment on the embedded RISC processor have been performed, and its result shows that the proposed method is more efficient than the case of using fixed ISA model.

Ⅰ. 서론

최근의 시스템들은 점차 복잡해지는 경향을 보이고 있다. 그래서, 많은 시스템들이 ASIC과 같은 하드웨어 부분과 RISC 프로세서와 같은 소프트웨어 부분으로 나뉘어져 구성된다. 또한, 칩의 집적도가 증가하면서 이러한 시스템을 단일 칩으로 구성하려는 반면에 제품을 개발하는 기간은 줄어드는 경향을 보이고 있다. 그러므로, 단일 칩에 포함된 하드웨어와 소프트웨어 부분을 통합 검증하는 문제가 개발 기간을 줄이는데 있어 중요한 걸림돌 중 하나로 인식되고있다. 그래서, 하드웨어와 소프트웨어의 코시뮬레이션(co-simulation)에 대한 연구가 많이 진행되어왔다[1][6][7].

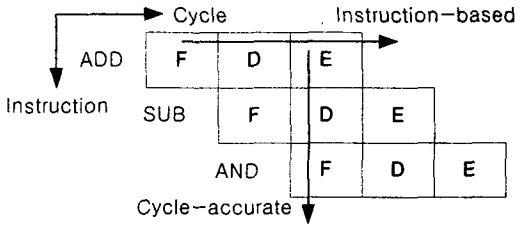
코시뮬레이션은 타겟 프로세서의 ISA (Instruction Set Architecture) 모델에서 실행되는 사용자 응용 프로그램 코드를 나머지 하드웨어 시스템과 동시에 검증한다. 일반적으로, 통합 검증에 사용되는 프로세서 모델은 instruction-based model이었다[4]. 이를 인터페이스 핀에 cycle-accurate한 타이밍을 제공하는 BIM(Bus Interface Model)과 결합하여 하드웨어 시뮬레이터와 연결하게된다. 반면에, [3]에서는 DSP 프로세서를 완전한 사이클 단위의 정확도를 가지도록 동작을 모델링하여 코시뮬레이션을 수행하였다.

본 논문에서는 시뮬레이션 구간에 따라 instruction-based model과 cycle-accurate model을 바꾸어 실행시킴으로써 원하는 구간에는 자세한 모델링을 제공하면서도 전체 시뮬레이션 시간을 줄일 수 있는 코시뮬레이션 방법을 제안한다.

Ⅱ. 코시뮬레이션 환경

2.1 ISA model

제안된 HW/SW 코시뮬레이션에서는 소프트웨어 부분을 시뮬레이션하기 위해 ISA 모델을 사용한다. ISA 모델은 타겟 프로세서의 명령어 집합의 기능적인 동작만을 모델링하는 것을 말한다. 이것은 프로세서의 HDL 모델처럼 내부의 타이밍 정보를 고려하지 않으므로 더 빠르게 실행될 수 있다. ISA 모델은 instruction-based model과 cycle-accurate model을 사용한다. 두 모델을 그림1에 표시하였다. 그림 1은 fetch, decode, execution의 3 stage pipeline을 가지고 있는 프로세서의 동작을 나타낸 것이다.

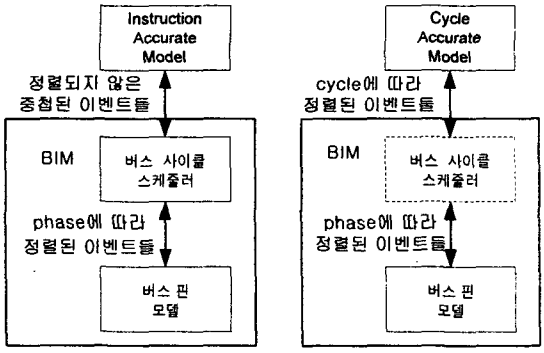


< 그림 1 > Pipeline operation

Instruction-based model은 타이밍을 고려하지 않고 명령어의 정확도를 가지게 프로세서를 모델링한다. 따라서, 이 모델은 명령어 경계에서의 프로세서 상태를 제공한다. 하지만, 이 모델은 최근의 프로세서들이 가지고 있는 중요한 특징 중의 하나인 파이프라인을 완전히 모델링하지는 않고, 모델이 한 번 실행될 때, 한 명령어를 그림1과 같이 파이프라인을 따라 여러 사이클의 동작을 시뮬레이션한다. Instruction-based model의 성능은 초당 2000에서 20000개의 명령어를 수행한다고 알려져 있다.[2] 이와는 다르게, 모델을 사이클 단위의 정확도를 가지도록 구성할 수도 있다. 이 cycle-accurate model은 한 번 실행될 때, 그림1에서처럼 파이프라인의 세로 축 부분을 시뮬레이션하게 된다. 모델이 동작할 때 파이프라인의 stall[5]이나 flush/refill 등을 모델링하므로 instruction- based model보다 더 느리게 실행된다. 두 경우의 모델 모두 memory, memory mapped I/O, I/O 등과의 연결을 위해서는 이들과는 phase의 정확도로 상호 작용하는 것이 필요하다.

2.2 BIM

ISA 모델은 C program으로 구성되기 때문에 하드웨어 시뮬레이터와 상호 작용하기 위해서는 중간에 Bus Interface Model이 필요하다. 이 모델 자체는 크게 버스 스케줄러와 버스 핀 모델의 두 부분으로 구성된다. 버스 스케줄러는 ISA 모델에서 발생된 읽기, 쓰기, 신호 할당과 같은 소프트웨어 이벤트들에 대한 정보를 모은 후에, 각각 대응되는 하드웨어 이벤트를 찾는다. 그리고, 각 클락 사이클에 올바른 순서대로 스케줄링한다. 이 트랜잭션들을 인터럽트나 메모리 지연 상태(memory wait state)와 같은 외부 입력들을 적절히 고려하기 위한 프로세서 구조에 대한 정보와 사이클 수에 대한 정보를 가지고 이미 pending하고 있는 트랜잭션들과 함께 스케줄한다. 한편, 버스 핀 모델은 하드웨어 시뮬레이터에 pin delay back- annotation을 제공한다.



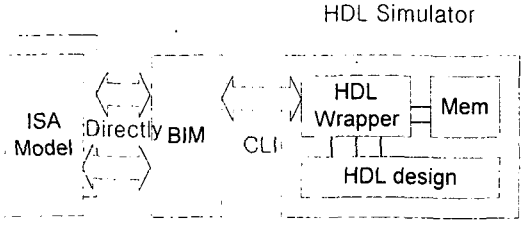
< 그림 2 > 모델 구조

그림 2에 instruction-based model과 cycle-accurate model에 대해 가능한 BIM들을 보여주고 있다. cycle-accurate model은 프로세서를 사이클 단위로 동작을 모델링하므로 BIM이 instruction-based model의 BIM보다 간단해진다. 사이클에 따라 정렬된 이벤트를 phase에 따라 정렬된 하드웨어 이벤트로 옮기기만 하면 되기 때문이다. 명령어의 중첩된 실행과 다른 파이프라인의 영향 등은 이미 모델 자체에서 다루어졌다.

2.3. Interface

코시뮬레이션에서 사용하는 ISA 모델은 두 가지 형태로 존재하게 된다. 하나는 실행 파일 형태로 존재하는 것이고 다른 하나는 오브젝트 파일 형태로 존재하는 것이다. 독립적인 실행 파일로 존재하는 경우는 UNIX IPC(Inter Process Communication)을 사용하여 하드웨어 시뮬레이터와 통신을 하게된다. 하지만, 이로 인한 오버헤드가 시뮬레이션을 수행하는 데 bottleneck이 된다. 오브젝트 파일로 존재하는 경우는 하드웨어 시뮬레이터에 의해 불려질 때마다 실행된다.

C program이 VHDL 디자인과 연결될 수 있는 것은 VHDL '93에서 정의된 Foreign attribute를 통해서이다. 하지만, 시뮬레이터 툴마다 다른 환경을 제공하는데 Synopsys VSS CLI, Mentor MTI FLI등이 있다.

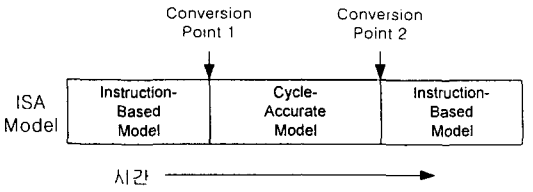


< 그림 3 > 전체 블록도

그림 3은 제안한 코시뮬레이션 환경의 전체 블록도를 나타낸 것이다. ISA model이 BIM와 결합되어 하드웨어 시뮬레이터에 연결되어 있다. ISA model과 같이 BIM도 오브젝트 파일 형태로 존재하며, 매 클락 사이클마다 하드웨어 시뮬레이터에 의해 불려진다.

Ⅲ. 모델 변환에 의한 코시뮬레이션

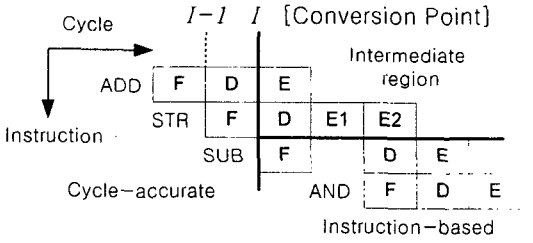
Cycle-accurate model은 프로세서의 동작을 좀 더 자세히 모델링하므로 instruction-based model보다 디버깅 기능을 더 제공한다. 하지만, 통합 시뮬레이션에 필요한 테스트 벡터의 길이가 길어지면 instruction-based model과의 실행 시간차가 점차 길어지게 된다. 즉, 모델링 정확도와 실행 시간에는 서로 trade-off 관계가 있는 것이다. 그런데, 대부분의 경우 항상 프로세서를 자세히 모델링할 필요는 없다. 그러므로, 이미 검증된 부분에 대해서 가능하면 간단한 모델로 실행시키면 전체 시뮬레이션 시간을 단축시킨다. 그림 4에서 시뮬레이션을 수행하는 동안 ISA 모델을 변환하는 것을 보여주고 있다.



< 그림 4 > 모델 변환

실행되는 ISA 모델의 변환은 타겟 프로세서에서 실행될 프로그램 코드나 특정 클락 사이클에 미리 변환 지점(Conversion Point)을 정해 놓은 후, 시뮬레이션을 수행시킴으로써 이루어진다.

두 모델 사이에 변환이 있을 때는 즉시 변하지는 않고 그림 5에서와 같이 중간 지역이 두 모델 사이에 존재한다.



< 그림 5 > 파이프라인에서의 모델 변환 과정

위 그림은 cycle-accurate model을 실행하다가 SUB 명령어부터는 instruction-based model을 실행하는 과정을 보여주고 있다. 변환 지점은 *I* cycle로 정하거나 SUB 명령어로 정하게 된다. *I-1* cycle까지는 cycle-accurate model을 실행하였다가 변환 지점인 *I* cycle에서는 intermediate region을 수행한다. 이 기간에는 instruction-based model과 유사하게 가로축 방향으로 실행되나, cycle-accurate model에서 수행한 부분은 수행하지 않는다. SUB 명령어부터는 완전한 instruction-based model이 실행된다.

Cycle-accurate model에서 instruction-based model로 변환되는 것도 위에서 설명한 것과 비슷하게 이루어진다. 단지 중간 지역에서 실행될 때 cycle-accurate model과 유사하다는 차이가 있을 뿐이다. intermediate region에서는 변환 지점이후부터 실행될

cycle-accurate model이 정확히 동작할 수 있도록 파이프라인의 상태를 적절하게 갱신한다.

Ⅳ. 실험

타겟 프로세서는 ARM 7을 택하여 ISA model을 cycle-accurate model과 instruction-based model로 ANSI C programming language로 구현하였다[5]. 하드웨어의 시뮬레이션은 상용 하드웨어 시뮬레이터를 통하여 이루어지도록 하였다. 실험은 몇 가지의 다른 계산 복잡도(computational complexity)와 I/O activity를 가진 예제들에 대해 수행되었다.

표 1에서는 cycle-accurate model만을 실행한 경우와 변환 지점을 instruction-based model과 cycle-accurate model이 50%씩 실행 되도록 설정한 hybrid model에 각 예제를 실행한 결과를 보여주고 있다. Simple은 낮은 계산 복잡도를 가진 예제이고, Complex는 높은 계산 복잡도를 가진 예제이다.

< 표 1 > 50%로 나누어진 경우

예제	Speed(Instruction Per Second)/Speedup	
	Cycle-accurate model only	Hybrid model
Simple	66K / 1	74K / 1.11
Complex	62K / 1	68K / 1.10

표 2에서는 hybrid model에서 각 instruction-based model과 cycle-accurate model의 비가 각각 25%, 50%,75%가 되도록 하여 각 예제를 실행한 결과를 보여주고 있다.

< 표 2 > 각 모델의 비가 다른 경우

예제	Instruction-based : Cycle-accurate		
	1 : 3	1 : 1	3 : 1
Simple	70K	74K	78K
Complex	65K	68K	72K

위의 실험 결과로부터 제안된 방법에 의해 시뮬레이션을 수행할 경우 cycle-accurate model만을 적용한 경우보다 최고 1.11배의 성능 향상을 얻을 수 있음을 알 수 있다.

Ⅴ. 결론

본 논문에서는 통합 시뮬레이션을 수행하는 구간에 따라 정확도가 다른 ISA 모델을 실행시켜서 정확도/디버깅 기능과 실행 속도에서 최적 화된 통합 검증 방법을 제시하였다. 이는 시뮬레이션을 시작하기 전에 변환 지점을 미리 설정해 놓음으로써 이루어지는 것이다. ARM 프로세서의 ISA 모델을 통한 실험은 특정한 시뮬레이션 구간에서는 프로세서의 cycle-accurate modeling과 자세한 디버깅 기능을 제공하면서 전체 시뮬레이션 시간을 줄일 수 있음을 확인하였다. 앞으로의 과제는 변환 지점을 결정함에 있어 최적화 된 지점을 찾는 것이다.

참고 문헌

[1] T. Albrecht, J. Notbauer, S. Rohringer, "HW/SW CoVerification Performance Estimation & Benchmark for a 24 Embedded RISC Core Design", DAC, pp.808-811, 1998

[2] J. Rowson, "HW/SW co-simulation", DAC. pp.439-440. 1994

[3] L. Guerra, J. Fitzner, D. Talukdar, C. Schlæger, B. Tabbara, V. Zivojnovic, "Cycle and Phase Accurate DSP Modeling and Integration for HW/SW Co-Verification", DAC, pp.964-969, 1999

[4] R. Earnshaw, L. Smith, K. Welton, "Challenges i n cross-development", IEEE Micro, pp.28-36, Jul y/Aug. 1997

[5] S. Furber, "ARM System Architecture", Addison Wesley, 1996

[6] R. Klein, "Miami: A hardware software co-simul ation environment", IEEE Int'l workshop on Rapi d System Prototyping, pp.173-177, 1996

[7] B. Lin, K. Van Rompaey, S. Vercauteren, D. Ver kest, I. Bolsens, H. De Man, "Designing single c hip systems" , ASIC, 1996