

내장형시스템을 위한 HW/SW 통합검증 환경 연구

김남도, 양세양
부산대학교 컴퓨터공학과
e-mail : ndkim@hyowon.pusan.ac.kr

Study on HW/SW Co-verification Methods for Embedded Systems

Nam-Do Kim, Sei-Yang Yang
Dept. of Computer Engineering, Pusan National University

요 약

최근 휴대폰단말기, PDA 와 같은 내장형시스템에 필수적으로 사용되고 있는 SoC(System On a Chip)에 대한 설계에서는 HW/SW 동시설계를 통한 설계생산성 향상이 필수적이다. 이에 따라서 설계검증에서도 HW/SW 통합검증의 중요성이 매우 커지고 있다. 본 논문에서는 이와 같이 내장형시스템을 위한 HW/SW 통합검증을 효율적으로 수행 할 수 있는 방법들인 co-simulation 과 co-emulation 및 co-prototyping 에 대하여 이들 방법들의 장단점과 더불어 이들을 통합한 새로운 검증방법인 집적 동시-검증(integrated co-verification) 기법에 대하여 논하기로 한다.

1. 서론

휴대폰, PDA 와 같은 소형화 및 저전력화가 중요한 내장형시스템(embedded system)설계에서 SOC(System-On-a-Chip)기술은 매우 중요하다. 최근의 내장형시스템 설계의 일반적인 추세는 HW 와 SW 를 동시에 설계하는 HW/SW 동시 설계(co-design)방법이 많이 이용되고 있다. 이와 같은 HW 와 SW 가 동시에 존재하는 시스템수준 설계 과정은 크게 시스템수준 설계생성 단계와 시스템수준 설계검증 단계로 나눌 수 있다. 설계기술과 집적도의 발전으로 점점 복잡해지는 시스템수준의 시스템설계에서는 전체 시스템수준 설계 시간과 비용의 70%를 시스템수준 설계검증에 투입하여야만 하는 심각한 시스템수준 설계검증위기 상황을 초래하고 있어 시스템수준 설계검증 생산성을 높일 수 있는 효과적인 방법론이 제시되지 않는다면 가까운 미래에 시스템수준 검증을 위하여 투입되어야 하는 비용과 시간은 더욱 크게 증가할 것으로 예상되고 있다[1]. 현재까지 시스템수준 설계검증에 보편적으로 사용되어지고 있는 동시-시뮬레이션(co-simulation)은 사용의 편의성, 강력한 디버깅 지원, 유연성, 매우 빠른 컴파일 속도 등의 매우 중요한 장점들이 있지만, 최근의 수백만 게이트급 이상의 시스템수준 설계검증에서부터는 매

우 낮은 검증속도와 인서킷(in-circuit) 능력의 결여 등의 문제점으로 이를 이용하여 설계검증을 성공적으로 수행하는 것은 극히 현실적이지 못한 것으로 인식되고 있다[2, 3, 4, 5]. 반면에 동시-에뮬레이션(co-emulation)은 고속의 검증속도와 인서킷 능력의 보유 등의 장점들이 있으나, 디버깅의 어려움, 사용의 어려움, 매우 긴 컴파일 시간, 높은 가격과 유지비용 등의 문제점들을 가지고 있다[3, 4, 5, 6].

본 논문에서는 동시-시뮬레이션과 동시-에뮬레이션 및 동시-프로토타이핑의 장단점들을 분석하고 이 방법들을 개방형 환경에서 강하게 결합시킨 새로운 시스템수준 검증 방법인 집적검증(Integrated Verification)방법을 제시한다. 제시하는 검증방식은 동시-시뮬레이션과 동시-에뮬레이션의 장점들만을 모두 취함으로써 고속의 검증 실행속도, 편의성, 강력한 디버깅 지원, 인서킷 능력, 유연성 등의 장점들을 매우 낮은 비용으로서도 실현할 수 있다. 따라서 이와 같은 집적검증 방법의 이용은 시스템수준 검증 생산성을 크게 높일 수 있으며 검증위기를 해결할 수 있는 강력한 해법 중의 하나가 될 수 있다.

2. 기존의 시스템수준의 설계검증 방법들

2.1. 동시-시뮬레이션(Co-simulation)

동시-시물레이션은 시스템수준 검증 대상을 소프트웨어적으로 모델링하고 고속의 프로세서를 이용하여 수행시킴으로 검증하는 것이다. 이 방법은 검증 대상시스템을 소프트웨어로 모델링하여 순차적으로 실행하여 검증하므로 복잡한 시스템검증에 사용하기에는 너무 느린 검증속도를 나타낸다. 또한 타겟 환경을 소프트웨어적으로 올바르게 모델링하는 것은 아주 어려우며 결과적으로 시스템수준 설계검증 비용과 시간을 증가시키는 결과를 초래하게 된다. 만일 인서킷 기능을 이용할 수 있다면 타겟 환경을 직접 연결함으로써 별도의 모델링 과정없이 정확한 시스템수준 검증환경을 이용하는 것이 가능하나 동시-시물레이션만으로는 이와 같은 인서킷 기능을 제공하지 못한다. 그러나 시스템수준 설계검증 대상의 모든 신호(선), 변수들에 대하여 재컴파일 과정 없이도 탐침이 가능함으로써 매우 편리한 디버깅 환경을 제공한다. 또한 HW 부분에 대해서는 함수적 검증뿐만 아니라 정확한 타이밍 모델링을 통하여 타이밍 검증까지도 가능하고, 다른 검증 방법들에 비하여 매우 빠른 컴파일 시간을 제공한다.

2.2. 동시-에뮬레이션(Co-emulation)

시물레이션 방식의 검증의 매우 느린 검증 속도를 하드웨어를 이용하여 해결하고자 하는 노력은 시물레이션 가속기(simulation accelerator)와 ASIC 에뮬레이터를 시장에 출현 시켰다[3, 4]. 과거의 ASIC 에뮬레이션은 테이프아웃하기에 앞서 최종 시스템수준 검증을 수행하는 수단으로서 사용이 되었으나, 현재는 시스템수준 설계 초기 단계에서의 결함들을 신속히 바로잡아 시스템수준 설계 시간과 비용을 감소시키기 위하여 사용하는 추세로 빠르게 변화되고 있다. 최근에는 이와 같은 요구사항을 반영하기 위하여 보편적으로 레지스터전송수준에서의 에뮬레이션이 가능하며, 최근 들어서는 행위적 수준의 에뮬레이션으로까지 확대되고 있다[6]. 이와 같은 에뮬레이션 기법을 시스템수준까지 확대한 것이 동시-에뮬레이션이다. 하지만 기술의 특성상 디버깅 환경이 유연하고 강력하지 못하고 에뮬레이션을 위한 컴파일시간이 매우 오래 걸린다. 또한 해당 시스템을 셀업하는 데에도 오랜 시간이 요구되며 시스템의 구입, 운영, 유지에 매우 큰 비용이 소요되는 것이다.

2.3. 동시-프로토타이핑(Co-prototyping)

시스템 설계·검증 엔지니어가 직접 시스템수준 검증 대상이 되는 시스템 구조에 제일 적합한 방식으로 프로토타이핑 시스템을 구성함으로써 프로토타이핑에 소요되는 비용을 최소화하면서 프로토타이핑의 성능은 최대화시킬 수 있다는 것이 제일 큰 장점이다. 프로토타이핑에 광범위하게 사용되는 FPGA 기반의 에뮬레이터의 수행 속도가 평균 2-3MHz 정도인 반면에 프로토타이핑에서는 최소 10MHz 에서 평균

20-40MHz 정도의 수행 속도를 어렵지 않게 이룰 수 있으며 특별한 경우에는 100MHz 이상의 수행 속도를 얻는 것도 가능하다. 하지만 프로토타이핑은 별도의 계측장비들을 추가적으로 마련하여야 하며, 이를 이용하여 효과적인 디버깅 환경을 구축, 사용하는 데에 많은 시간과 비용이 투입되어야만 한다. 이와 같은 문제는 특히 FPGA 를 사용한 동시-프로토타이핑의 경우에 특히 심각하다. 탐침을 위한 빈번한 배선 배치에 많은 시간이 소모되며, 최근의 고집적 FPGA 는 PGA, BGA, fine-grained BGA 방식의 팩키징 방식을 사용하여 계측기로서 탐침하기가 힘들거나 불가능하기 때문이다. 또한 프로토타이핑 플랫폼을 자체적으로 구성하는데에 많은 시간과 관련 지식이 필요하다. 이와는 별도로, 프로토타이핑 플랫폼을 자체적으로 구성하는데에 많은 시간과 관련 지식이 필요하다는 것이 프로토타이핑의 또 다른 단점이 될 수 있으나, 최근들어서 이를 지원하는 다양한 툴들 뿐만 아니라 semi-custom 형식의 프로토타이핑 플랫폼들도 여러 벤더들로부터 제공되고 있어 예전에 비해서 훨씬 빠른 시간 안에 프로토타이핑 환경을 갖출 수 있다[7, 8].

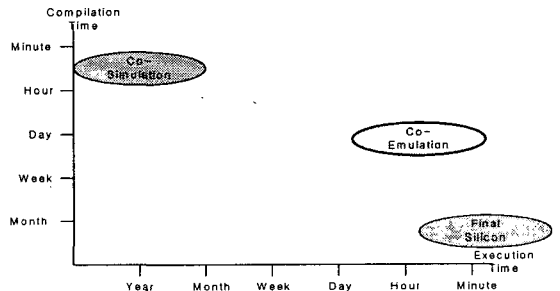


그림 1. 동시-시물레이션과 동시-에뮬레이션의 수행시간/컴파일시간 비교

표 1. 동시-시물레이션/동시-에뮬레이션/동시-프로토타이핑의 장단점 비교

	Cost	Accuracy	Co-Simulation	Flexibility	Debugging Features
Co-Simulation	---	---	+++	+++	+++
Co-Emulation	+	++	---	-	-
Co-Prototyping	+++	+++	-	---	---

3. 집적 동시-검증(Integrated Co-Verification)

그림 1 과 표 1 에서와 같이 2 장에서 기술한 방법들은 각각 다른 독특한 장점과 약점들을 동시에 가지고 있다.

본 논문에서는 완전 개방된 방식에 기반 하여 "임의의" 동시-시물레이션 기술(동시-시물레이터)과 "임의의" 동시-에뮬레이션 기술(동시-에뮬레이터), 또

는 "임의의" 동시-시뮬레이션 기술(동시-시뮬레이터)과 "임의의" 동시-프로토타이핑 기술(동시-프로토타이퍼)을 강하게 결합시킨 집적 동시-검증 기술을 제안하고, 다양한 장점들에 대하여 언급한다. 그림 1. 동시-시뮬레이션과 동시-에뮬레이션의 수행시간/컴파일시간 비교

표 1. 동시-시뮬레이션/동시-에뮬레이션/동시-프로토타이핑의 장단점 비교

제안되는 방법은 임의의 동시-에뮬레이터와 임의의 동시-시뮬레이터를 연결시키거나, 혹은 임의의 동시-프로토타이퍼와 임의의 동시-시뮬레이터를 연결시키기 위하여 시스템수준 동시-검증 브리지(co-verification bridge)를 사용한다(그림 2). 이와 같은 시스템수준 동시-검증 브리지를 통하여 시스템수준 설계 검증이 되는 대상은 시스템수준 검증 플랫폼이 되는 동시-에뮬레이터, 동시-프로토타이퍼, 또는 동시-시뮬레이터에서 수행되는 시스템수준 설계 검증 대상의 모델들 간에 정보를 고속으로 교환할 수 있도록 하는 것이 가능해짐으로 다른 시스템수준 설계 검증 방식들간의 실시간 전환(동시-에뮬레이션에서 동시-시뮬레이션, 또는 그 반대이거나 동시-프로토타이핑에서 동시-시뮬레이션 또는 그 반대로)을 전환 횟수에 아무 제한 없이 가능하게 한다. 이와 같은 자유로운 실시간 전환을 통하여 집적 동시-검증 방식은 다음과 같은 효과적이며 신속한 시스템수준 검증을 가능하게 하는 기능을 제공할 수 있다.

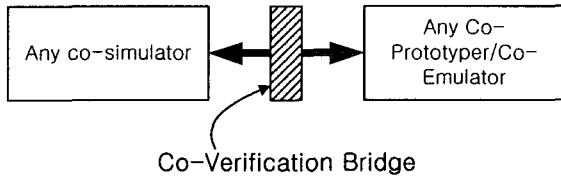


그림 2. 집적 동시-검증 시스템의 구성

3.1. 고속의 시스템수준 검증 실행속도

시스템수준 검증수행의 특정 기간 동안에 매우 빠른 검증 실행속도를 얻고자 하는 경우에는 동시-에뮬레이터나 동시-프로토타이퍼 상에 하드웨어적으로 모델링된 시스템수준 검증 대상을 수행시킴으로 손쉽게 이를 달성할 수 있다. 만일 이에 앞서서 동시-시뮬레이션이 수행되고 있었다 하면, 동시-검증 브리지를 통하여 동시-시뮬레이터의 최종 수행정보를 동시-에뮬레이터나 동시-프로토타이퍼로 실시간 이동시킴으로 연속되어지는 시스템수준 검증을 매우 빠르게 수행하는 것이 가능하다. 특히, 동시-프로토타이핑을 이용하는 경우에는 시스템수준 검증 실행속도를 적어도 20 MHz 이상 되도록 하는 것이 쉽게 가능함으로 초고속의 시스템수준 설계 시스템수준 검증이 가능해진다.

3.2. 디버깅 능력과 사용의 편의성

디버깅 과정에서 수많은 신호선들 또는 변수들의 값을 일정기간동안 탐침하고자 하는 경우에는 동시-시뮬레이터의 제공하는 디버깅 기능을 이용하여 검증 대상에 존재하는 어떠한 신호선들이나 변수들에 대하여 신속한 탐침이 가능하다. 만일 현재의 시스템수준 검증 수행이 동시-에뮬레이터나 동시-프로토타이퍼에서 수행되어지고 있다면, 동시-검증 브리지를 통하여 동시-에뮬레이터나 동시-프로토타이퍼의 최종 수행정보를 동시-시뮬레이터로 실시간 이동시킨 후, 이를 탐침을 원하는 신호선들이나 변수들에 대하여 탐침이 가능하다. 또한 제안된 복합 시스템수준 검증 기술은 시스템수준 설계·검증 엔지니어의 입장에서 현재 사용중인 검증 방법을 그대로 사용할 수 있도록 하면서도 현 검증 방법에 의하여 제기되고 있는 문제점들을 해결할 수 있는 새로운 검증 방법을 손쉽게 통합하는 것을 가능하도록 한다. 따라서 시스템수준 검증을 수행하는 설계·검증 엔지니어는 시스템수준 검증 환경을 전혀 바꾸지 않거나, 환경변화를 최소화하면서도 검증 생산성을 획기적으로 높이는 것이 가능해진다.

3.4. 인서킷 능력과 낮은 검증비용

제안되는 검증 방식은 동시-프로토타이핑이나 동시-에뮬레이션과 같은 하드웨어 기반의 시스템수준 검증 시스템을 이용할 수 있음으로 인서킷 능력을 가지게 되며 이를 활용하면 타겟 환경으로부터 입출력을 집적 공급받고 공급해줄 수 있음으로 실제 타겟 환경에서의 실동작(real operation)의 시스템수준 검증이 가능하다. 또한 제안되는 집적 동시-검증 방식을 채용하기 위하여서 시스템수준 설계·검증 엔지니어는 현재 사용하고 있는 시스템수준 검증 환경 및 검증 시스템을 모두 바뀌어야 하는 것이 아니며, 현재의 시스템수준 검증 환경은 그대로 유지하며 사용 중인 검증 시스템과 새로운 검증 시스템을 통합한 집적 동시-검증 시스템을 구성하면 된다. 따라서 이와 같은 시스템수준 검증 환경 및 시스템을 매우 낮은 비용으로 구축 할 수가 있다. 더욱이 이와 같은 집적 동시-검증 기술을 사용하는 경우에는 동시-시뮬레이션, 동시-에뮬레이션, 동시-프로토타이핑을 각각 단독으로 사용하여 시스템수준 검증을 하는 경우에는 제공할 수조차 없는 다양한 기능을 제공할 수 있음으로 인하여 시스템수준 검증의 생산성을 획기적으로 높이는 것이 가능함으로서 검증에 소요되는 비용과 시간을 크게 낮추는 것이 가능하다.

4. 실험

본 논문에서 제안된 새로운 시스템수준 검증 방식인 집적 동시-검증 방법의 효능을 실험적으로 시스템수준 검증하기 위하여 저가격내장형시스템에 광범위하게 사용되고 있는 마이크로컨트롤러의 설계에 집적 동시-검증 방법을 적용하여 검증을 수행하였다. 사

용된 복합 시스템수준 검증 환경의 구성은 다음과 같다. 시뮬레이터는 Model Technology 사의 Vsim 5.4 Windows-NT 버전으로 구성하였고 프로토타이퍼는 자체적으로 제작한 FPGA 보드(Xilinx Virtex XCV50-PQ240 장착)로 구성하였으며, 검증 브리지는 시뮬레이터가 인스톨된 펜티엄 III PC의 PCI 슬롯을 통하여 이 둘이 연결되어지게 하였다.

4.1. 하드웨어 디버깅

타이밍 시스템수준 검증 능력을 점검하기 위하여 마이크로컨트롤러 시스템수준 설계에서 의도적으로 게이트되어진 클럭(gated clock)을 국지적으로 사용하였다(그림 3). 이와 같은 게이트되어진 클럭을 갖는 회로는 클럭 스큐의 정도와 두 레지스터들간의 신호 전달 시간의 정도에 따라 multi-stepping 이 일어날 수 있는 위험성을 내포하고 있다. 본 시스템수준 설계에서는 의도적으로 이와 같은 multi-stepping 이 일어날 수 있도록 회로를 시스템수준에서 설계하였다. 시스템수준 검증의 시나리오는 우선 프로토타이퍼상의 FPGA 에 마이크로컨트롤러의 시스템수준 설계 비트파일을 다운로드하여 하드웨어 모델링을 완료한 후에 테스트 프로그램을 수행시키면서 게이트 활성화(gate enable) 신호를 생성하는 명령어 직전까지 하드웨어 기반의 함수적 시스템수준 검증을 고속으로 수행한 다음, 이 시점에서 시스템수준 검증 브리지를 통하여 프로토타이퍼의 최종 수행정보를 Vsim 으로 실시간 이동시켰다. 그 다음에는 Vsim 에서 게이트 활성화 신호를 생성하는 명령어의 머신사이클들을 정확한 타이밍 모델을 이용하여 수행시킴으로서 multi-stepping 이 일어나는 상황을 발견할 수가 있었다.

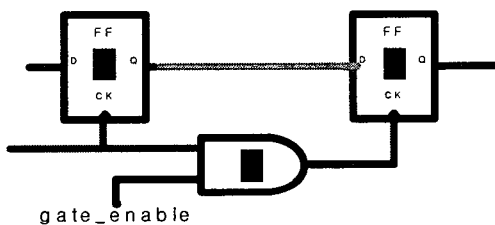


그림 3. Multi-stepping 을 야기할 수 있는 게이트되어진 클럭

4.2. 소프트웨어 디버깅

고속의 시스템수준 검증 실행 속도와 디버깅의 편의성을 실증하기 위하여 간단하나 오랜 시간의 수행을 요하는 64 비트 카운터 어셈블리 프로그램을 실행시키면서 시스템수준 검증을 수행하였다. 어셈블리 코드에는 64 비트의 특정 값(0xE0000000)이 되었을 때 1 을 증가하는 대신에 2 를 증가시키는 어러코드를 고의적으로 삽입하였다. 이와 같은 실험에서도 시스템수준 검증의 시나리오는 우선 프로토타이퍼 상의

FPGA 에 마이크로컨트롤러의 시스템수준 설계 비트파일을 다운로드하여 하드웨어 모델링을 완료한 후에 테스트 프로그램을 수행시키면서 8 개의 8 비트 레지스터로 구성되는 64 비트 값이 0xE0000000 가 될 때까지 하드웨어 기반의 함수적 시스템수준 검증을 고속으로 수행한 다음, 이 시점에서 시스템수준 검증 브리지를 통하여 프로토타이퍼의 최종 수행정보를 Vsim 으로 실시간 이동시켰다. 그 다음에는 Vsim 에서 회로에 존재하는 모든 신호선들에 자유로운 탐침을 수행함으로써 인스트럭션 메모리에 존재하는 어러 코드를 발견할 수 있었다.

5. 결론

본 논문에서는 시스템수준 설계 검증의 대표적인 방법들인 동시-시뮬레이션과 동시-에뮬레이션, 또는 동시-시뮬레이션과 동시-프로토타이핑을 "개방형" 환경에서 강하게 결합시킨 새로운 시스템수준 집적 동시-검증 방법을 제시하였다. 이는 소프트웨어 기반의 검증 방법과 하드웨어 기반의 검증 방법의 장점들만을 수용하여 매우 낮은 비용으로도 고속의 검증 실행속도, 강력한 디버깅, 인서킷 능력 등의 장점들을 얻을 수 있다. 특히, 시스템수준 설계·검증 엔지니어는 현재의 시스템수준 검증 환경을 그대로 유지하면서 사용 중인 시스템수준 검증 시스템과 새로운 시스템수준 검증 시스템을 통합한 집적 동시-검증 시스템을 구성하면 된다. 따라서 이와 같은 집적 동시-검증 방법의 이용은 최소한의 시간과 비용을 투입하면서도 시스템수준 설계검증 생산성을 크게 높일 수 있음으로 인하여 내장형시스템을 위한 시스템수준의 검증위기를 해결할 수 있는 매우 강력한 해법 중의 하나가 될 수 있다.

참고문헌

[1] Richard Foster, "A Design Style to Simplify IP Integration and Verification," White paper, VLSI Technology, Inc.(<http://www.vlsi.com>), 1999.
 [2] Voyager Datasheet, IKOS(<http://www.ikos.com>), 2000.
 [3] J. Babb, R. Tessier, M. Dahl, S. Hanano, D. Hoki, and A. Agarwal, "Logic Emulation with Virtual Wires," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, June 1997
 [4] Mercury Datasheet, Quickturn(<http://www.quickturn.com>), 2000
 [5] MP4 Datasheet, Aptix(<http://www.aptix.com>), 2000
 [6] N. Kim, H. Choi, S. Lee, S. Lee, I. Park, and C. Kyung, "Virtual Chip: Making Functional Models Work on Real Target Systems," in *Proc. of DAC*, 1998.
 [7] Certify Datasheet, Synplicity(<http://www.synplicity.com>), 2000
 [8] FPGA Stuffer Datasheet, SpeedGate(<http://www.speedgateinc.com>) 2000

본 연구는 소프트웨어진흥원의 휴대형 멀티미디어 단말기 연구센터관련 연구비 지원에 의한 연구결과입니다.