



Virtual Prototype for Early SW bring up & HW/SW co-verification

I. 서론

전통적인 SW에서 임베디드(Embedded) SW로의 변형되는 현재 Trend와 Multi-Core Architecture가 이끄는 디지털 기기의 내부 표준 변화는 임베디드 SW의 복잡성과 개발시간의 증가를 요구하고 있는 상황입니다. 이 Multi-Core 혁명을 통해 새로운 프로그래밍 언어, HW / SW 종속성, HW팀과 SW 팀을 위한 새로운 개발도구와 새로운 Work Flow의 Need가 발생하며, 현재 시장을 선도하는 반도체 공급 업체에서는 HW개발을 시작하기 전에, SW로 개발을 가속화하여 제품의 품질과 Warranty를 보장하고, 고객 수를 크게 증가 할 수 있도록 Virtual Prototype을 사용하고 있다.

II. 본론

1.1. What is a virtual prototype?

Virtual Prototype은 호스트 컴퓨터 에서 실행되는HW 시스템 의 실행 가능한 SW 모델로서, Virtual Prototype과 SW 관점에서 바이너리 호환의 핵심인 CPU 명령어 세트, 메모리 맵 디코딩, 레지스터, 인터럽트, HW 블록 등등의 SW와 관련이 있는 HW를 시뮬레이션 하여 ROM 또는 펌웨어 코드, 운영 체제 (OS) 및 AUTOSAR 같



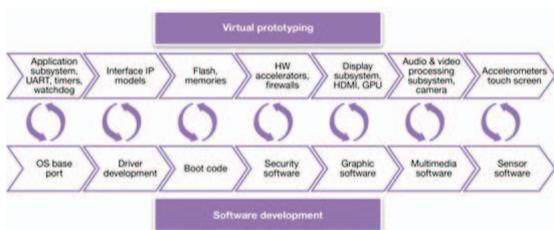
김 형 주
Product Solution
Specialist / Synopsys



은 상위 레벨 SW, 미들웨어 및 응용 프로그램을 포함하여, 전체 SW 이미지를 실행 할 수 있게 한다. Virtual Prototype은 SW 개발자가 실제 개발 보드를 사용하는 것과 유사한 방식으로 이진 프로그램 코드가 로드 된 상태에서 실행 및 디버깅 할 수 있는 타겟 시스템의 기능을 제공한다. 그러나 아래에서 논의 되는 바와 같이, Virtual Prototype은 물리적 관점에서 실제 HW와 몇 가지 주요 차이점 이 있다.

Virtual Prototype은 HW Clock cycle 정확도와 낮은 수준의 HW 동기화와 같은 세부 사항을 조율하여 기존의 실제 HW 중심의 에뮬레이터와 시뮬레이터에 비해 매우 빠른 성능으로 실행이 가능하다. 또한 SW의 호환성을 위해 시스템의 범위와 복잡성에 따라 적절한 모델링을 하며, 일반적으로 실제 HW 디자인에 비해 적은 개발 시간과 더 적은 리소스를 필요로 하는 점이 주요 장점이다.

더 현실적인 Virtual Prototype을 만들기 위해서 Main 시스템 외에 UART, USB, 이더넷 과 같은 I/O 인터페이스를 포함하는 칩의 I/O를 가상화 할 수 있다. 예를 들어, USB 디바이스 드라이버를 설치하면, Virtual Prototype은 새로운 멀티 미디어 플레이어 또는 스마트 폰과 같은 실제 USB 장치로 호스트 컴퓨터에 나타날 수 있고, 이는 I/O 테스트 및 기존의 디버깅을 지원하기 위해 Virtual Prototype과 완벽하게 상용되는 디버거 솔루션의 인터페이스에 대한 고급 시스템 시나리오를 테스트 할 수 있다.

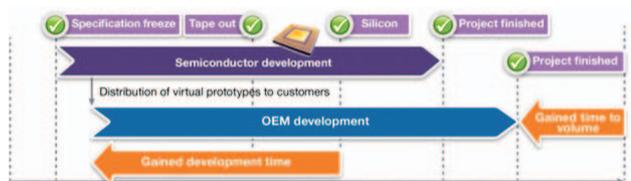


〈그림 1〉 Software stack requirements drive the virtual prototype staging

Virtual Prototype의 또 다른 장점은 현재 개발 중인 SW Stack의 Subset에 관련된 시스템의 일부를 generation하여 SW 개발을 Step별로 진행할 수 있다는 점이다. F-1에서 보이는 바와 같이 개발자는 개발 중에 SW 스택의 서브 시스템이 기능면에서 점진적으로 증가함에 따라 대응이 가능하며, 특정 SW 개발 단계에 필요한 HW를 확인 할 수 있으며. 또한 가상의 아키텍처를 통해 SW 개발을 쉽게 진행 할 수 있다. 예를 들면, 시스템 온 칩 (SoC) 에서 새로운 칩에 SW를 이식하게 되면, OS 뿐만 아니라 새로운 CPU 코어에 포팅이 되어야 하기 때문에 단계적으로 Virtual Prototype에서는 첫 번째 CPU 코어를 업그레이드하여 새로운 칩에 대한 SW개발을 할 수 있게 한다. 포팅을 쉽고 간단하게 하는 장점이다.

1.2. Virtual prototypes in embedded software development

Virtual Prototype은 (〈그림 2〉 참조) SW 개발을 기존의 방법보다 일찍 시작하고 새로운 제품을 개발 하고 테스트 하는데 필요한 총 시간을 단축하며, HW 개발과 동시에 진행 할 수 있다. FPGA Prototype또는 기타 실제 HW보드에 의존하고 있는SW 개발자의 관점에서 볼 때는 Virtual Prototype이 HW/SW 통합의 의미로 개발 및 디버깅이 훨씬 더 쉬어지며, 더 편리한 SW 개발로 생산성에 직접적인 영향을 줄 수 있으며, 개발 비용을 절감하며, 전체적인 SW 개발의 Resource와 시간을 대폭 줄이고, 추후 발생할 수 있는 제품의 기능상



〈그림 2〉 Early delivery of virtual prototypes means early delivery of finished products

의 문제위험을 감소시킨다.

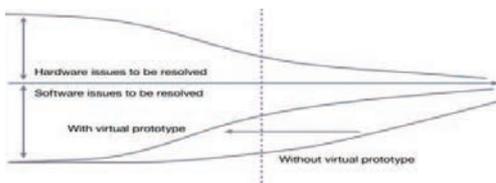
물론 HW 디자이너의 관점에서는 HW function/Timing검증을 위해 FPGA 프로토타입과 함께 사용할 수 있는 HW개발 관점의 Prototype도 필요하다.

개발자들이 실리콘 또는 물리적 HW의 가용성을 기준으로 90-95% 수준의 SW 기능의 달성을 원한다면, Virtual Prototype을 이용하여 미리 9-12개월 전에 개발을 시작할 수 있다. 이는 HW 설계와 병렬적으로 SW 개발 및 디버깅을 동시에 가능하게 하는 효과를 갖는다.

이후 HW 설계/검증이 준비가 되면 많은 SW 스택을 사용할 수 있으므로 HW 에뮬레이션 및 FPGA 기반 프로토타이핑과 같은 HW 검증 기술을 사용하여보다 광범위한 테스트도 가능하게 되며, 실제로 Virtual Prototype을 사용하고 있는 많은 개발자는 이러한 개발 과정의 혁명으로 전체 제품 개발시간이 2~3개월 이상 단축이 되고 있다고 보고하고 있다. 또한 HW/SW 통합으로 생산성 향상 및, 가상 시제품까지 해결 될 수 있다. 이러한 부분은 직접 비즈니스에 영향을 미치게 된다. 또한 많은 개발자가 3주 만에 3~6개월이 소요되는 새로운 실리콘의 제작 및 Delivery 시간의 감소를 통해 높은 제품 품질과 개발 일정 생산 Risk를 줄여주므로 비즈니스 수익성에 큰 영향을 준다.

1.3. Virtual prototypes in embedded software debugging

SW 디버그 및 테스트는 2012년 기준으로 전체적인 반도체 설계 시간의 23%를 차지하고 있고 SoC의



〈그림 3〉 Virtual prototypes enable early convergence of software quality

**Virtual Prototype 기법은
임베디드 소프트웨어의 개발,
특히 디버깅 프로세스를 효율적으로
수행할 수 있는 장점이 있다.**

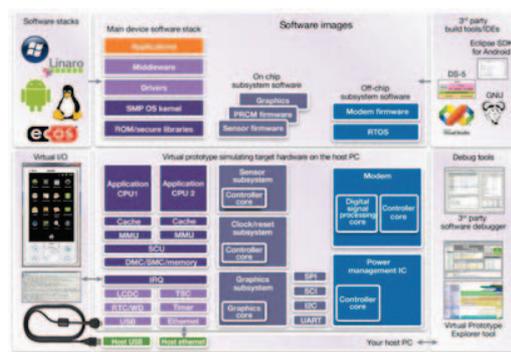
사이즈 및 복잡성이 꾸준히 증가하고 있기에 임베디드 SW 개발의 대부분의 시간과 자원을 소모하는 단계이다. 그래서 새로운 HW 아키텍처와 새로운 SW의 스택에 발생하는 버그를 해결하는 것이 Embedded SW개발의 중요 과제이다. Virtual Prototype의 조기 가용성은 〈그림 3〉에 도시된 바와 같이, 개발 및 테스트에서, HW 설계와 병렬로 진행할 수 있는 9-12개월간의 시간을 제공하며, 이와 같이 일찍 Test를 진행할수록 전체적인 SW 품질에 명확한 효과가 나타난다. 결과적으로, 전체적인 제품의 품질은 높아지고 제품을 출하를 빨리 할 수 있다.

Virtual Prototype을 이용한 임베디드 SW의 디버깅 프로세스를 통해 다음과 같은 절차로 SW 개발 및 디버깅을 진행할 수 있습니다.

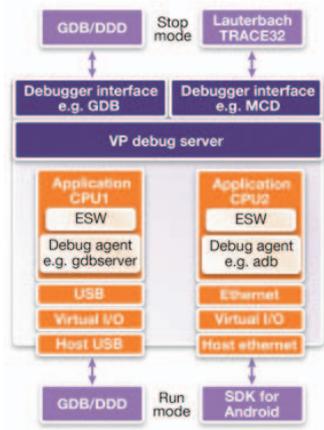
- 결함을 발견
- 증상을 통해 결함을 관찰
- 근본 원인을 분석
- 시나리오를 재현
- 문제 해결
- 결함이 사라진 것을 증명하기 위해 업데이트된 코드로 시나리오를 다시 실행

1.3.1. Software tool integration

〈그림 4〉에서 알 수 있듯이, Virtual Prototype은



〈그림 4〉 Virtual prototype-based software development environment



〈그림 5〉 A virtual prototype in “stop” mode and “run” mode

실제 HW 와 동일한 플랫폼을 통해 필요한 모든 포트와 대상 장치의 SW 구성 요소를 디버깅하여 완전한 임베디드 SW 개발 환경을 제공한다.

또 하나의 중요한 SW개발환경의 요소는 표준 임베디드 SW 개발 도구 및 기존 워크 플로우와의 통합부분이다. 구체적으로 예를 들면, 업계 표준의 SW 디버거 솔루션 (ARM DS-5, 라우터 바흐 TRACE32, GNU gdb등)을 이 Virtual Prototype과 원활하게 통합한다면, ARM의 신규 SW 개발 도구를 학습 할 필요가 없으며, SW 에 대한 변경 사항에 대해서 업무를 줄 일 수 있다.

효과적인 디버그 시나리오 지원을 통해 “정지 모드”와 “실행 모드” (각각 〈그림 5〉의 상단과 하단 참조)의 방식으로 SW 디버그 환경을 제공한다.

전자는 가상의 환경에서 디버그 서버인 MCD 같은 산업 표준 인터페이스를 디버거에 연결하여 물리적 대상인 JTAG 연결과 유사한 타겟에서 저레벨 디버그 제어를 허용하고, 후자는 “실제 Real I/O” 활용하여 장치 드라이버와 호스트 컴퓨터의 물리적 인터페이스를 활용하여 실제 세계에 가상 I/O 를 연결 할 수 있는 Virtual Prototype 의 기능이다. 실제 USB 인터페이스는 실제 물리적 USB 인터페이스로 디버거에 나타나지만, Virtual Prototype은 “실행” 모드에서 디버거를 물리적으로 연결하는 것이 필요하지 않다.

결과적으로, 임베디드 SW 개발팀은 성공적으로 실제

HW 타겟에서 SW개발 및 디버깅을 Virtual Prototype 으로 이동하여 디버깅하여 빠른 결정과 가시성 그리고 제어성을 가질 수 있다.

1.3.2. Visibility

디버깅을 위해 Virtual Prototype을 사용하는 또 다른 큰 장점은 SW 코드, HW 블록, 서브시스템 및 실제 HW를 이용 시 확인이 어려운 시스템 전체를 개별 Block으로 나누어 진행할 수 있다는 점이다. 시뮬레이션 속도뿐만 아니라 시인성이 장점으로 잘 알려져 있고, 실행 이후에 재생 문제에서의 control이 가능하다는 것이 장점이다.

Virtual Prototype에서HW의 register level을 확인하며, HW와 SW의 상호 작용 문제를 디버깅 할 수도 있으며, 디버그 서버 덕분에 시뮬레이터와 연결된 모든 SW 디버거를 연결하는 Virtual Prototype구성이 가능하며, 시뮬레이션 시간 이 정지되더라도 타이머 틱이 없고, 시간이 초과되지도 않는다. 이러한 관점에서 SW 개발자는HW 및 SW 의 visibility 확보를 통해 보다 안전하고 일관성 있는 결과를 얻을 수 있으며, 메모리, 레지스터는 물론 CPU신호, 주변 장치, 서브 시스템, 전력 관리 블록, 센서 등의 모든 경계에 걸쳐 관찰이 가능하다.

Virtual Prototype 사용자는 Virtual Prototype이 제공하는 향상된 가시성, 제어 능력으로 버그를 쉽게 찾을 수 있고, 추가적으로 OS 및 드라이버의 버그를 분리 할 수 있다.

또한, Virtual Prototype사용자 설문 조사를 통해서 확인된 바로는 동일한 크기의 SW 팀과 비교 시 Virtual Prototype 사용자의 경우 같은 Resource에서 더 많은 일을 할 수 있으며 그리고 프로젝트 당 더 적은 자원을 필요로 하여2-5배의 개발 및 디버깅 생산성 향상을 가능하게 한다.

1.3.3. Control

Virtual Prototype을 통해 SW개발 및 디버깅을 진행할 때, 사용자는 전체 시스템을 중지하고 메모리, 레



지스터의 내용을 수정하거나 다른 작업에 영향을 미치는 일이 없이 인터럽트를 설정하고 그 시점에서부터 계속작업을 할 수 있다.

1.3.4. Determinism

항상 동일한 환경에서 동일한 결과를 생성하는 특성의 Virtual Prototype의 SW 디버그 환경은 매우 유용하며, 이를 통해 사용자에게 결함을 수정하는 데 걸리는 시간단축을 제공하며, 결함 시나리오를 재현 할 수 있는 기능을 제공한다. 만약 이러한 기능이 없다면, SW 디버깅은 장시간이 소요되는 작업이다. 이는 예를 들면, 시나리오에서의 이벤트 시퀀스와 Arrive time등을 예측할 때 나타나는 버그를 처리 할 때 또한 어려운 작업이다.

1.3.5. Multicore debug

사용자가 실제 시스템의 특정 프로세서 코어를 중지 하면 어떻게 될까? 시스템의 나머지 정지 할 수 없기 때문에 시스템의 상태는 즉시 손상 될 수 있다. 시스템에 여러 개의 프로세서가 있는 경우에 특정 프로세서 코어가 중단되는 상태가 되면 사용자는 불안할 수 있다. 하지만 모든 것을 중지시키더라도 Virtual Prototype을 이용하면 사용자에게 아무것도 영향을 미치는 것 없이 시스템으로 접속 할 수 있다. 이는 모든 잠금 단계에서 실제 data이동 없이 프로세서와 HW의 전체 시스템 상태를 추적 할 수 있기 때문이다. HW와 SW그리고 다중 프로세서를 포함 하더라도 임의로 중단점을 설정할 수 있다. 또한 Virtual Prototype은 Cache Coherence 관련된 복잡한 문제에 멀티 코어 아키텍처의 추가를 쉽게 할 수 있도록 멀티 스레드 및 멀티 프로세스 등을 지원한다.

1.3.6. Ease of deployment

실제 HW 의 경우 개별적인 SW 개발자에서 1:1공급하기에는 비용 및 시간 등의 문제가 발생할 수 있다.

특히 장거리로 분산되어 있는 SW 개발팀에 제공하기 위해서는 cost & delivery time측면에서 어려운 점이 있다.

Virtual Prototype의 경우 일종의 SW개념으로 실행 가능한 파일로 만들어 일반적인 SW 개발 툴 과 유사한 비용 수준에서 큰 숫자로 복제 및 전달이 가능하다. 편리하고 간단하게 E-mail또는 내부 인트라넷을 통해 추가 SW 팀 에 Virtual Prototype 설치 이미지를 전달 및 재배포 함으로서, 큰 비용 및 Delivery time의 손실 없이 각각의 SW team에서 SW개발 작업이 가능하게 한다.

1.3.7. Fault injection

Virtual Prototype은 SoC 또는 시스템의SW 모델이기 때문에 레지스터 및 온칩 또는 시스템의 구성 요소의 신호 라인 값을 용이하게 제어 할 수 있다. 이는 GUI 또는 스크립트를 통해 레지스터의 값을 설정하여 특정 상황(Fault Injection)을 만들어 특정 조건에서 시스템의 동작을 테스트 할 수 있다.

잘못된 설정이나 입력에 대한 시스템의 반응을 테스트할 때 특히 편리하다. 이를 통해 자동차

제어 기능과 같은 중요 애플리케이션을 위한 높은 품질의 SW를 만들 수 있다.

Virtual Prototype 기법은 모델링을 통해 더 빠른 개발 단계에서 더 빠른 속도로 소프트웨어 및 하드웨어의 테스트를 진행할 수 있게 한다.

1.4. Validation and verification: test early, test often

임베디드 시스템의 복잡성은 더 많은 테스트를 더 빠른 시점에 필요로 하고 있다. Virtual Prototype의 경우는 modeling을 통해 더 빠른 시점에 더 빠른 속도로 SW 개발 및 디버깅을 가능하게 한다.

1.4.1. Scripting software tests

Script based SW test를 통해 많은 Senario에서 Script를 통해 Senario, SW및 HW의 Configuration을 손쉽게 변경해 가며 SW 테스트를 수행할 수 있다.



또한 Virtual Prototype에서 임베디드 SW는UART 직렬연결을 통해 가상의 테스트 인터페이스를 사용할 수 있다.

또한, Virtual Prototype 의 “가상 I/O” 기능을 사용하여 키패드, LCD 및 터치 스크린 등을 포함한 I/O장치를 Virtual Prototype의 사용자 인터페이스로서, 호스트 PC, 스크립팅 또는 일반적인 SW GUI 테스트 도구에GUI 로서 실현되는 플랫폼과의 사용자 인터페이스를 자동화 하는데 사용될 수 있다.

이를 통한 이점은 다음과 같다.

- 더 나은 테스트를 실행하여 높은 품질획득 -가상 환경 시스템의 모든 측면 에서 우수한 제어성 및 가시성을 제공하기 때문에, 시스템에 어려운 코너 케이스를 넣어 더 복잡한 시나리오를 테스트하는 것이 가능하다.
- 효율적 테스트를 통한 비용 절감 - 전체 시스템이 가상 이고 데스크톱 PC 및 서버 팜에서 모두 실행할 수 있기 때문에 테스트를 쉽게 설정 하고 자동화 할 수 있습니다. 예를 들어, Failure테스트에서 SW 버그를 기존의 블랙 박스 기술을 이용하는 것 보다 더 빠르고 효율적으로 할 수 있다.
- 확장성을 통한 테스트 비용절감-simulation 을 통해 다양한 조건에서 테스트시스템 1,000 개를 구축 할 수 있다 이를 통해 비용 절감과 다양한 test 조건을 충족시킬 수 있다.

1.4.2. Preparing post-silicon validation test software

일반적으로 반도체의 동작검증을 위해 새로운 실리콘이 의도한 대로 작동하는지 확인하기 위해 진단 테스트 방법의 광범위한 세트를 개발하고 다시 반도체 공급 업체 내부의 팹, 포스트 실리콘 검증팀 에서 새로운 실리콘의 유효성을 검사 한다. 이러한 flow는 전형적인 칩

내부에 다양한 코어에서 실행되는 SW 테스트를 통해 진행된다. 지금까지 테스트 환경의 개발은 일반적으로 초기 물리적 프로토 타입 에서 수행 또는 실제 실리콘이 가능하게 된 경우에만 가능했다.

이러한 검증 테스트 스위트의 전달이 반도체 공급 업체 및 고객에게는 새로운 실리콘을 출하 구동하기 위해 필요한 중요한 요소 중의 하나로서 만약 Virtual Prototype을 이용하게 되면 사용자가 효과적으로 “포스트 실리콘 개발”로의 이동뿐만 아니라 이 초기 검증 테스트의 진행을 가능케 하여 많은 도움을 줄 수 있다. 실제로Virtual Prototype을 사용 중인 반도체 업체는 제품개발을 위한 테스트가 초기단계에서 진행되고 시간이 단축 됐을 뿐만 아니라, 전체적인 제품의 버그 및 작동 상 문제도 감소하였다.

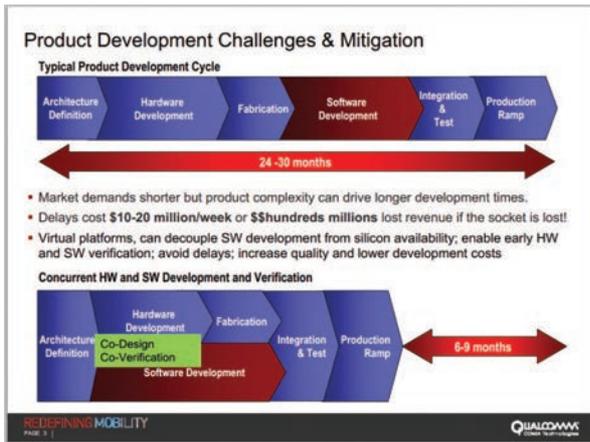
이와 더불어, 개발 시 전력 관리는 특히 적절한 실리콘 검증 을 보장하고 초기 SW 가용성 접근을 보증하기 위해 현재 설계에서 중요한 고려 사항 중 하나이다. 소비 전력은 신규 SoC 의 유효성에 매우 중요한 요소이며, 실제SoC 의 특정 HW 구성 요소 에 의해서만 구동되는 것이 아니라, 더 많은 전체 애플

오늘날 모든 반도체 회사는 매우 복잡한 시스템, 실리콘 가용성 및 제품 개발 완료 시간의 증가의 어려움에 직면해 있으며 이를 해결하기 위한 새로운 도구와 방법으로 Virtual Prototype 기법이 해답이 될 수 있다.

리케이션 SW 스택에서 사용되고 파워 관리 SW 에 의해 제어되며, 실제 응용 SW 및 전력 관리 SW를 실행하는 실제 시나리오의 맥락에서 적절한 검증 없이는 실제 전력 / 성능 Trade off의 결과를 얻을 수 없다. Virtual Prototype의 경우 SW Scenario based Power analysis 및 전력관련 SW개발 디버깅을 초기에 진행 할 수 있기 때문에 많은 도움이 될 수 있다.

1.4.3. Software-driven hardware verification and early hardware/software integration

HW의 초기 Function 모델 을 갖는 것은 SW 개발의 가속화는 물론Virtual Prototype 에 대한 더 많은 사용 case를 열어 많은 측면에서 혜택을 제공한다. 개발 중



〈그림 6〉 Customer success story

인 시스템에 대한 Virtual Prototype을 위한 정확한 Function모델은, HW 플랫폼의 golden reference 및 RTL 검증 흐름에서 사용할 수 있기 때문에 단순 HW에 편향된 검증뿐만 아니라 HW/SW가 결합된 상태에서의 통합된 검증환경을 제공한다.

Use Case

- 초기 Test case (테스트 벤치) 개발 : RTL 검증팀이 “시스템 통합 테스트” 개발을 가속화하기 위해 초기 Virtual Prototype의 장점을 적용할 수 있다. 이 테스트는 일반적으로 제한된 Random 테스트와 같은 IP 중심의 검증 방법론을 보완하기 위해 SoC 및 IP 통합 문제 검증에 초점을 둔 “임베디드 SW 테스트”이다. 이 flow에서 Virtual Prototype은 검증 초기에 사용할 수 있는 유용한 검증 도구이다.
- Virtual Prototype은 실제 RTL DUT가 없는 상황에서 사용된다. Virtual Prototype은 빠른 RTL 시뮬레이션보다 몇 배 이상의 시뮬레이션 속도를 제공하며, 모든 RTL DUT에 비해 높은 생산성 테스트 벤치 및 테스트 케이스 자체를 제공하며, 그리고 초기 개발을 할 수 있다.
- HW/SW 동시 검증 : 개별 IP 또는 서브 시스템의 RTL을 사용할 수 있게 되면 RTL 모델을 이용하여 RTL 시뮬레이터에서 사용할 수 있는 TLM/RTL 통합 시뮬레이션 기능을 사용하여

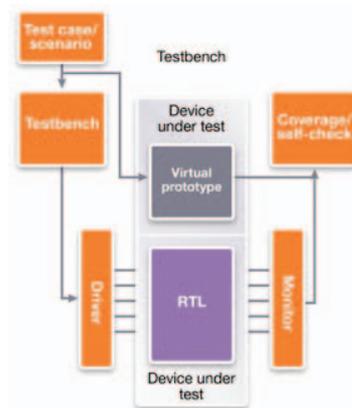
Virtual Prototype으로 통합할 수 있다. 이를 통해, RTL 서브 시스템은 고급 테스트 케이스로 사용할 수많은 SW 스택과 함께 전체 시스템의 환경에서 테스트할 수 있다. 뿐만 아니라 완성된 제품의 초기 HW/SW 검증을 수행하는 HW와 SW 팀 간의 업무에 유용한 협력 환경을 제공한다.

- TLM/RTL 등가 검사를 위해 Golden reference를 사용하여 검증팀은 일반적으로 RTL 구현이 검증된 상태에서 System Verilog 같은 언어를 이용하여 “Reference model”을 개발한다.

이를 통해 HW와 SW 간 동일 모델을 통해 불일치(다양한 소스로부터 유래되는 HW 에러, SW 에러, HW 버그 또는 버전의 차이, TLM 모델 문제 등)를 사전에 검출할 수 있다. 이러한 설계 Flow는 현재의 RTL 레벨에서 제기되는 “Top-Down” 설계 방법을 제공할 수 있는 필수 요소이며, 현재 디자인의 복잡성 증가와 HW & SW 통합 등으로 야기되는 설계, 검증시간의 증가 없이 현재 수준을 유지하는 중요한 기술이다.

1.5. Creating and managing virtual prototypes

1.5.1. When? The sooner the better



〈그림 7〉 TLM/RTL co-simulation: a virtual prototype is used as virtual “DUT” to develop RTL test cases while the RTL is developed



새로운 사용자에게 “Virtual Prototype의 장점”의 질문에 대한 가장 많은 대답은 “HW 설계에 병렬로 Virtual Prototype을 적용할 수 있다”는 것이다. 오늘날 모든 반도체 회사는 매우 복잡한 시스템, 실리콘 가용성 및 제품 delivery 시간증가의 어려움에 직면해 있고, 그들에게 시간 내에 작업을 수행 할 수 있도록 수 있는 새로운 도구와 방법에 대한 솔루션으로 Virtual Prototype이 해답이 될 수 있다.

초기 SW 개발을 가능하게 하는 Virtual Prototype 방법론을 채택할 경우 회사가 완전히 자신의 제품 계획에 따라 제품 개발을 실행 할 수 있는 장점을 극대화 할 수 있다. 이를 위해서는 초기에 HW 사양을 통해 Virtual Prototype을 구축하기 위해 모델의 초기 개발을 필요로 하며, 가상 시제품이 SW 개발팀의 요구에 따라 개발되어야 하므로, 그것은 HW 팀/모델링 팀과 SW 팀 간의 더 친밀한 협업을 필요로 하게 된다.

1.5.2. Building and using models

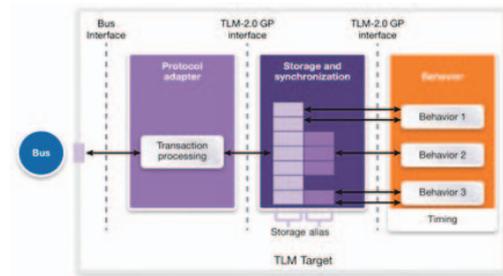
Virtual Prototype은 IEEE 1666-2011 표준 HW 모델링 언어인 SystemC 를 사용하여 트랜잭션레벨 모델 (TLM)로서 모델링되어 사용된다.

SW 동기화 이벤트를 활성화하기에 충분한 타이밍 정확도로 HW 기능을 모델링하고 빠르게 실행되며 실제 SW 스택을 디버깅 할 수 있는 Loosely timed 모델로 Virtual Prototype을 제작하게 된다.

- SystemC를 이용한 TLM - 2.0 modeling : 균일성 및 IP model의 상호 운용성을 촉진하기 위해 채택되었으며 Accellera 의 업계 표준이다. 통신 동작 및 타이밍의 분리를 고려한 코딩 스타일을 채택하고 특정 Virtual Prototype 유스 케이스의 속도 및 시인성의 요구 사항을 만족하기 위한 모델이다

<그림 8>의 다이어그램 모델은 각각의 모델이 분리되어 인터페이스를 구현 하도록 설계하는 방법을 보여 준다.

각 모델 구성 요소의 기능 분리로, 각각의 모델에서 구축된 Virtual Prototype 은 점진적으로 추가되거나

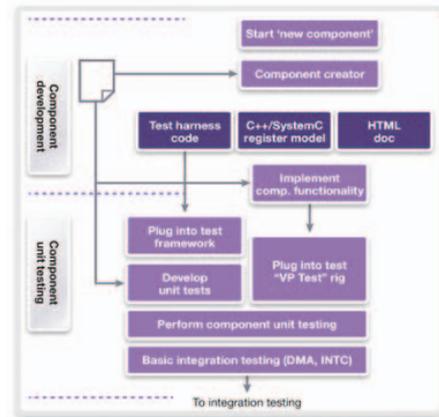


<그림 8> Target model coding style

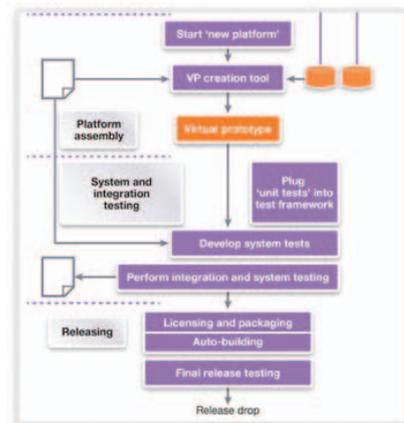
프로젝트의 요구와 작업흐름에 따라 작은 기능단위로 분석 할 수 있도록 모듈 형으로 creation될 수 있다

1.5.3. Leveraging test-driven development

Virtual Prototype에서 사용되는 모델의 품질을 보장하기 위해 그 Virtual Prototype 의 기초모델 생성과



<그림 9> Component Modeling Workflow



<그림 10> Virtual prototype modeling workflow



	2012	2013	2014	2015	CAGR
Americas	29.6	33.4	36.9	41.0	11.4%
Japan	11.6	12.8	13.5	14.9	8.9%
Rest of APAC	13.5	15.5	17.7	19.6	13.1%
EMEA	10.6	10.7	11.3	12.0	4.4%
TOTAL	65.3	72.3	79.4	87.5	10.3%

Source: VDC/Research, 2014

〈그림 11〉 Worldwide shipments of virtual prototyping solutions by geographic region, 2012–2015 (US \$M)

병행하여 테스트 프로그램을 개발하는 것이 중요하다. 모델은 단위 테스트 플랫폼에 통합되어야 하며, 임베디드 SW 테스트가 추가 되면 모델 기능을 발휘하도록 작성되어야 한다.

아래의 워크 플로우 다이어그램은 가상의 프로토타입 작성, 통합 테스트 및 출시의 단계 중 구성 요소 모델링 (〈그림 9〉, 〈그림 10〉)을 통해 Virtual Prototype 제작을 위한 가장 좋은 방법을 보여준다.

1.5.4. Market adoption and availability

〈그림 11〉은 전 세계적으로 걸쳐 Virtual Prototype 솔루션 도입에 대한 시장 조사 결과를 보여준다.

또한 앞으로의 Virtual Prototype 솔루션의 예상 성장 가능성을 보여 준다 .

트랜잭션 레벨 모델과 자신의 반도체 IP를 대표하는 라이브러리를 제공하는 ARM (주) 시놉시스 (Synopsys) 등이 대표적으로 Virtual Prototype을 지원하고 있는 업체이다.

III 결론

Virtual Prototype은 임베디드 SW 개발의 기존 플로우를 개선하여, HW/SW 동시 개발을 가능케 하여 전체 반도체 및 시스템개발 일정을 단축시키는 효과를 가져 올 수 있으며, 보다 엄격한 테스트 방법을 통해 제품의 Quality향상에 도움을 줄 수 있다.

참고 문헌

- [1] 2011 Embedded Software Engineering Market Technologies and Statistics, p.10, VDC
- [2] "ARM DS-5," accessed on March 3, 2014, <http://ds.arm.com/>
- [3] "Lauterbach TRACE32," accessed on March 3, 2014, <http://www.lauterbach.com/frames.html?home.html>
- [4] "GDB: The GNU Project Debugger," accessed on March 3, 2014, <http://www.sourceware.org/gdb/>
- [5] "SystemVerilog for Verification: A Guide to Learning the Testbench Language Features," Chris Spear, Greg Tumbush, Feb. 14, 2012, ISBN-13: 978-1461407140
- [6] "Accellera Systems Initiative," accessed on March 3, 2014, <http://www.accellera.org/downloads/standards/systemc/tlm>
- [7] "Accellera Systems Initiative," accessed on March 3, 2014, <http://www.accellera.org/downloads/standards/systemc/tlm>



김형주

- 한국 외국어 대학교 전자공학 학사
- 2005년~2010년 Davan Tech / Technical & Sales Support Manager
- 2010년~2011년 MAGMA Design Automation/ Product & Sales Manager
- 2011년~현재 Synopsys /Product Solution Specialist

〈관심분야〉
SoC Architecture Design, Emulation/FPGA Prototype, Virtual Prototype