

IP 검증을 위한 PCI 기반 리프로그래머블 설계 기능 에뮬레이션 환경 구현

최광재, 조용권, 이문기
연세 대학교 전기전자 공학과
전화 : 02-2123-4731 / 핸드폰 : 019-617-0251

A design of PCI-based reconfigurable verification environment for IP design

Kwang Jae Choi, Yong Kwon Cho, Lee Moon Key
Dept. of Electrical and Electronic Engineering, Yonsei University
E-mail : kjchoi@spark.yonsei.ac.kr

Abstract

The verification of software part and HW/SW interface suffer from the absence of the hardware platform at the end of partitioning and coding phase in design cycle. In this paper we present the design of easy verification for hardware design. Hardware and software engineer can verify their software program and hardware design for a chip that is emulated in proposed verification environment. Besides, designer can easily design the DEMO system.

I. 서론

최근 디지털 회로는 빠른 시장 상황의 변화와 반도체 집적도의 증가로 인해, 디자인 사이클은 짧아지고, 테스트에 걸리는 시간은 전체 회로 설계에서 많은 비중을 차지하게 되었다. 복잡한 회로를 어떻게 빨리 테스트를 하느냐가 시장에 먼저, 제품을 먼저 내놓을 수 있는 중요한 요소가 된 것이다. 최근 많은 EDA 툴 회사에서는 다양한 디자인 검증 환경을 위한 소프트웨어/하드웨어 Co-simulation, 하드웨어 시뮬레이션 Accelerator, 하드웨어 Emulator 등을 개발 공급하고 있으며, 소프트웨어 HDL simulator는 가장 일반적으

로 많이 사용되어 지는 툴이다. 그러나 복잡한 디자인을 시뮬레이션하고, 소프트웨어 Program 과 하드웨어를 Co-Simulation 하는 것은 고성능의 컴퓨터를 필요로 하게 되며 시그널의 이벤트 발생순서에 따라 순차적으로 시뮬레이션을 하게 되므로 매우 긴 시간이 걸리게 된다[1]. 또한 소프트웨어 시뮬레이션은 실제 상황의 일부분만을 관찰 가능하기 때문에 응용시스템을 적용하기 매우 어렵다.[2] 최근 많은 디자인은 여러 복잡한 IP를 사용하여 디자인하는 방법을 사용하고 있으므로 각각의 IP들이 하나의 시스템에 집적 되었을 때 나타날 수 있는 디자인 오류에 대해서 소프트웨어 시뮬레이션은 더욱 테스트하기 어려워진다.

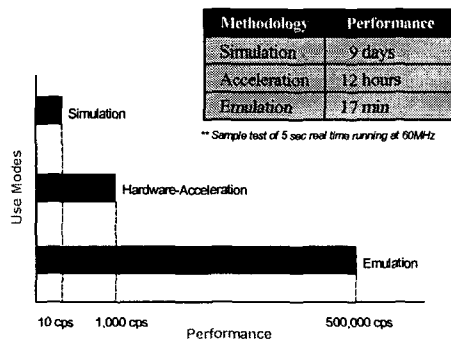


그림 3 Test Performance [3]

그러므로 보다 빠른 디자인 테스트를 하기 위해서는 실제 상황을 좀더 정확히 모델링 할 수 있는 환경과 빠른 테스트 속도(그림 1)가 무엇보다 중요하다. 이런 방법으로는 하드웨어 Acceleration, 에뮬레이션, Prototyping 이 있지만, Prototyping은 실제 하드웨어를 제작해야 하므로 제작 시간이 걸리고 다른 디자인에 재사용을 할 수 없다.[1] 그러므로, 본 논문에서는 재사용 가능하고 쉽게 사용할 수 있는 에뮬레이션 환경을 설계 제안한다.

EDA 회사에서 판매하는 하드웨어 Emulator [4]는 가격이 엄청나게 비쌌 뿐만 아니라, 사용법 또한 복잡하다. 하지만, 하드웨어 Emulator에서 많이 사용하고 있는 FPGA(Field programmable Gate Array)의 집적도가 매우 향상되어, 적은 비용으로 사용하기 충분한 디자인 검증환경을 설계가 가능하게 되었다. Emulator와 호스트 컴퓨터간의 데이터 전송 속도가 Emulator의 속도를 저하 시킬 수 있는 기존의 Emulator가 가지고 있는 단점[1]을, PCI를 사용하여, 보다 빠른 에뮬레이션 속도를 구현할 수 있었으며, PCI의 API를 이용하여 소프트웨어 Simulator 또는 소프트웨어 Model과 연결이 쉽게 할 수 있었다. 본 논문에서는 FPGA를 이용한 Design 에뮬레이션 Board를 설계하고, PCI API를 이용하여 C 모델과 같이 검증환경을 디자인 하였다.

II절에서 RVE(Reconfigurable Verification Environment)의 개요, III절에서는 RIC(Reconfigurable Internal Controller)구조, IV절에서는 디자인의 플로우, V절에서는 RVE 적용사례를 알아보고 VI 절에 결론을 맺는다.

II. 디자인 검증환경의 개요

검증환경은 FPGA, RIC, Memory Sub-system, PCI bridge, PCI driver, UI(User interface) 로 이루어져 있으면, 각각의 기능은 다음과 같다.

FPGA : Target design를 에뮬레이션 하기 위해 사용한다. 본 디자인에서는 Xilinx Vertex 1000E를 사용.

RIC : Emulator의 전체 시스템을 제어하기 위해 Internal controller를 CPLD(Complex Programmable Logic Device) 사용하여 이용하여 구현한 것임.

Memory Sub-system : Target Design에서 필요로 하는 ROM과 RAM를 지원하기 위해 FLASH ROM, SRAM, Configuration ROM 으로 구성됨.

UI : PCI API를 이용하여 UI를 설계.

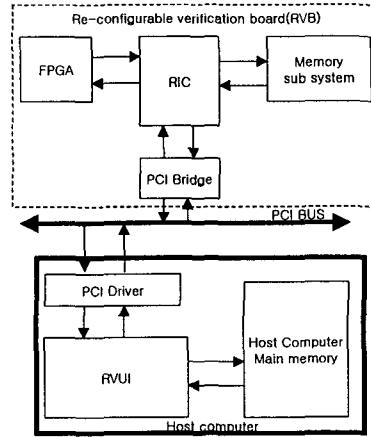


그림 4 검증환경 block diagram

UI는 PCI API를 사용하여 에뮬레이션 보드를 제어 가능하며, 여러 가지 프로그램과 함께 연동 할 수 있는 유연한 설계가 가능해졌다. RIC는 디자이너에게 Target FPGA와 RVB상의 Memory Sub-system, PCI bridge를 쉽게 연결 가능 할 수 있도록, 간단한 Interface를 제공한다. Memory Sub-system은 Target FPGA 내에 부족한, Memory 자원위해 외부 메모리를 지원하며 FPGA Configuration ROM를 가지고 있다.

III. RIC의 구조

RIC (Reconfigurable Internal controller)는 디자이너가 RVE에 자신의 디자인 쉽게 적용 할 수 있도록, SRAM 과 유사한 인터페이스 구조, Memory Mapped I/O, 32bit Scan Chain 기능을 제공하며, RVB상의 모든 제어신호와 데이터의 흐름을 제어 한다.

RIC내에 Memory buffer는 RIC 와 Target FPGA간의 Interface를 SRAM Interface와 유사한 구조를 지원하고,(그림 3) PCI bridge, Memory sub-system 사이에서 중간 버퍼 역할을 한다. 또한, FPGA에 Wait signal를 제어하거나, Target FPGA의 Main Clock를 제어 하므로써, PCI Read Delay 또는 Memory Read Delay를 제어한다.

Memory buffer 와 함께 Address Decoder는 (그림4)과 같은Memory map 구조를 가지고 있다. 이런 메모리 맵은 Target FPGA와 RVB 상의 Memory Sub-system, PCI bridge와의 복잡한 Interface를 단순화 시킬 수 있다. 32-bit Scan chain 은 컨트롤 신호의 초기화와, 원하는 곳에 신호의 변화를 관찰하기 위한 기능을 한다. 관찰하고자 하는 신호 선을 32-bit

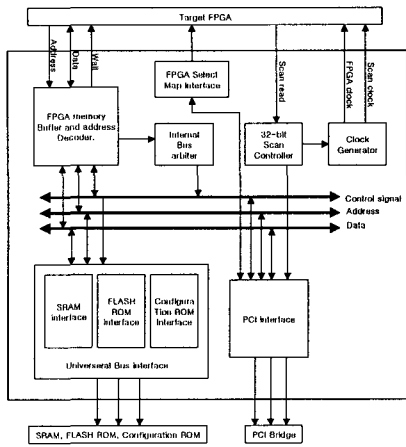


그림 5 RIC의 구조

Scan Cell과 연결하므로써, 클럭과 동기된 출력의 변화를 빠른 속도로 입출력 할 수 있다. 이 Scan Chain은 FPGA 내에 존재하는 JTAG를 사용하지 않고, 32-bit Scan cell를 합성 가능한 코드로 설계를 하였으며, 디자이너는 디자인의 원하는 부분의 HDL 상에서 쉽게 삽입 할 수 있다. 결과적으로, 이런 Scan Chain은 고정된 입력과 출력 포트 수를 줄일 수 있고, 넓은 버스 폭을 가지므로 빠른 속도의 데이터 전송이 가능하다. 32-bit Scan Cell를 5개 사용하고, Scan Chain의 동작 속도가 10Mhz 라면, 2Mhz의 속도로 전체 시스템을 테스트 할 수 있다. 이런 기능은 많은 테스트 벡터를 테스트하고, 결과를 얻는데 매우 유용한 기능이다.

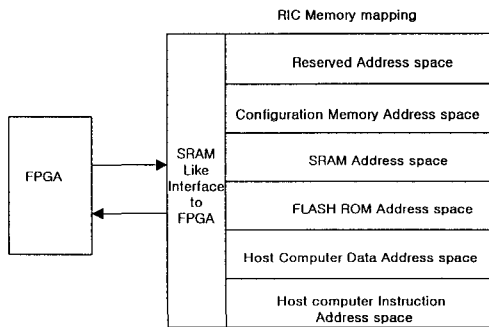


그림 6 RIC의 Memory Map

IV. 검증 과정

RVE를 일반 검증환경에 적용하는 것은 일반적인 FPGA Prototyping 하는 것과 비슷한 과정을 가진다.

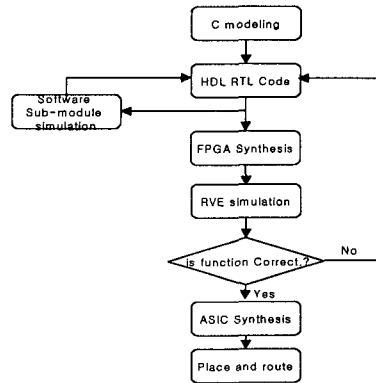


그림 7 디자인 과정

(그림 5). 하지만, RVE 에서는 디자이너에게 어떤 디자인에도 적용가능한 일관된 Prototyping 과정과 에뮬레이션 환경을 제공한다. 그러므로써, 소프트웨어 프로그래머와 하드웨어 디자이너가 전체 디자인 과정에서 좀 더 일찍 만나 디자인의 버그를 찾을 수 있는 환경을 제공 받을 수 있다.

V. RVE 적용사례

많은 Test vector를 빠른 시간동안 처리 할 수 있는 RVE를 MPEG2 영상 인코더의 중요한 블록 중 하나인 모션 벡터 추출기에 적용하였다. 2장의 352×288 픽셀의 영상을 입력 받아 2장의 영상 사이에 왜곡값과 움직임 변화를 모션 벡터로 출력하게 된다. (그림 6)

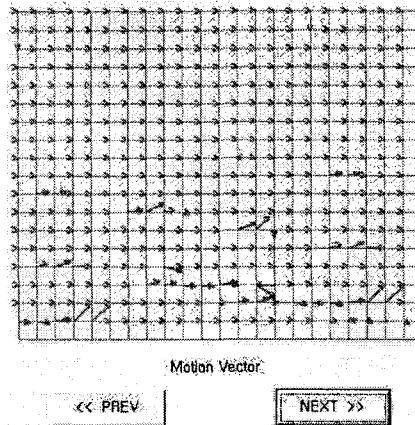


그림 8 모션 벡터 출력

이 블록 정합 움직임 추정기는 입력 테스트 벡터로

방대한 자료의 입력 데이터가 들어갈 뿐만 아니라, 계산량은 소프트웨어 HDL 시뮬레이션을 할 경우 1장의 영상 대한 모션 벡터(그림 6)를 추출하는데 메모리 1Gbyte RDRAM, Windows2000, 펜티엄 1.8 Ghz 에서 35분이 걸릴 정도로 방대 하다. HDL 소프트웨어 시뮬레이션 시간이 오래 걸리고, 많은 테스트 벡터를 필요로 하는 디자인인 경우 RVE에 적용하기에 가장 적합한 디자인이라 할 수 있다. 먼저, 각 영상의 프레임 데이터를 PCI를 통해 RBV(그림 7)로 보내고, 다시 PCI를 통해 모션 벡터를 받아 이전 프레임과 모션 벡터를 사용하여 C 언어 프로그램을 통해 영상을 재구성하는 방법으로 테스트를 하였다. 이런 방법은 계산량이 많은 블록 정합 움직임 추정기의 연산을 RVB의 Target FPGA에서 에뮬레이션 하므로 4 MHz의 속도로 테스트가 가능 했으며, 실시간으로 초당 30프레임의 모션 벡터를 추출 할 수 있었다.

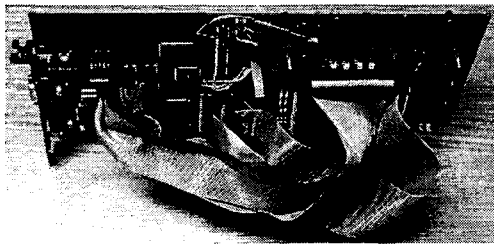


그림 9 RVB(Reconfigurable Verification Board)

VI. 결론

디지털회로의 디자인의 크기가 커지므로 서, 테스트가 전체 디자인 과정에서 차지하는 비중이 매우 커졌다. 그러므로 전체 디자인 설계 시간을 줄이기 위해서는 디자인 테스트 시간을 줄이는 것이 중요하다.

본 논문에서 제안된 PCI 와 FPGA를 이용한 디자인 에뮬레이션은 고성능 컴퓨터에서 하는 소프트웨어 시뮬레이션보다 매우 빠른 속도를 보였다. 그리고 소프트웨어와 하드웨어간의 인터페이스를 PCI API를 통해 설계할 수 있어, 보다 다양한 응용이 가능하다.

제안된 RVE 는HDL 디자인에서부터 에뮬레이션 테스트까지 간단한 과정을 가지고 있다. 이런 장점은 하드웨어 디자이너와 소프트웨어 디자이너가 전체 시스템 디자인 과정에서 좀 더 일찍 만나 하드웨어/소프트웨어 Co-Simulation 할 수 있으며, 반복되는 테스트와 버그 수정 사이클이 짧아 전체적인 테스트 크게 줄일

수 있다.

ACKNOWLEDGMENT

본 논문에서 사용한 EDA Tool은 IDEC를 통해 제공 받았다.

참고 문헌

- [1] Fast simulation of HDL models *Skold, S.; Ayani, R. IEEE Potentials* , Volume: 14 Issue: 5 , Dec. 1995-Jan. 1996 Page(s): 14 17
- [2] SRAM형 FPGA를 이용한 Rapid Prototyper 개발 대한 전자공학회지 23호 1996. 11
- [3] IKOS Performace test www.ioks.com
- [4] Highly configurable control boards: a tool and a design experience *De La Torre, E.; Riesgo, T.; Uceda, J.; Macip, E.; Rizzi, M. Rapid System Prototyping, 2000. RSP 2000. Proceedings. 11th International Workshop on* , 2000 Page(s): 174 179
- [5] Xilinx 3.1 Foundation development system reference guide. www.xilinx.com
- [6] PLX 9054 PCI bridge Data sheet www.plxtech.com