

논문 2006-43SD-10-7

# IEEE 1149.1 테스트 기능이 내장된 PCI/USB 통합 인터페이스 회로의 설계

(Design of PCI/USB Interface Controller with  
IEEE 1149.1 Test Function)

김영훈\*, 김기태\*, 박성주\*\*

(Younghun Kim, Kitae Kim, and Sungju Park)

## 요약

IEEE 1149.1 바운더리 스캔 칩들로 구성된 보드를 테스트하기 위한 패턴은 반드시 비트 스트림으로 구성되어야 한다. 그러나 이러한 비트 스트림을 생성하는 일은 IEEE 1149.1 표준에 대한 완벽한 지식이 필요하므로, 전문지식이 없는 SoC설계자에게는 상당히 어려운 일이다. 본 논문에서는 Test Ready PCI 와 Test Ready USB로 정의한 PCI와 USB 장치를 통해 편리하게 테스트를 수행할 수 있게 도와주는 테스트 인터페이스 컨트롤러를 제안한다. 이 제어기는 TI사와 Lucent사에서 명령어 단위의 수준에서 테스트 비트 스트림을 생성 하기위해 개발한 테스트 버스 컨트롤러를 기반으로 하여 테스트 전문 지식이 없는 설계자도 쉽게 테스트 패턴을 생성하여 테스트를 수행할 수 있는 장점이 있다.

## Abstract

In order to test the board with IEEE 1149.1 boundary scan design, the test sequence must be applied as the bit stream. However it is very tedious job to generate the test bit sequence since it requires the complete knowledge about the 1149.1. This paper introduces a convenient PCI/USB interface controller, named as Test-Ready PCI (TRPCI) and Test-Ready USB (TRUSB). Test Bus Controller has been developed by TI and Lucent aiming to generate the test bit stream as an instruction level, thus even the novice test engineer can easily generate the test sequence.

**Keywords :** IEEE 1149.1, test bus controller, TRUSB, TRPCI

## I. 서론

보드 수준에서의 테스트는, 보드에 실장된 칩 내부의 테스트 및 칩과 칩 사이의 상호연결선에 대한 고장 유무를 확인한다. 그러나 보드 설계 기술의 발달로 보드 테스트는 매우 어려운 문제가 되었고, 테스트 용이

성을 위해 IEEE 1149.1 표준이 제정되었다<sup>[1]</sup>.

IEEE 1149.1 표준은 테스트 패턴의 인가 및 결과 관측 과정을 비트 단위로 제어해야 하는 어려움이 있다. 이러한 문제점을 해결하여 효과적인 테스트를 하기 위한 몇 가지 테스트 제어기가 제안되었다<sup>[2, 3, 4]</sup>.

PCI 버스는 주로 고속의 인터페이스를 요구하는 회로에 사용되고 있고, 특히 PC는 PCI 버스에 기초를 두고 있다. 또한 USB 장치 역시 PC에 기초를 둔 장치로서 간편한 방법으로 PC와 고속 시리얼 통신이 가능하다. 따라서 현재 PC와의 고속 인터페이스 구현을 위해서는 PCI 버스 및 USB 장치의 사용이 필수로 요구된다. 본 논문은 다음과 같이 구성 되어 있다. II장에서는 보드 테스트를 위한 표준인 IEEE 1149.1과 기존의 테스트

\* 정회원, 한양대학교 컴퓨터공학과  
(Dept. of Computer Science & Engineering,  
Hanyang University)

\*\* 정회원, 한양대학교 전자 컴퓨터 공학과  
(Dept. of Electronical Engineering Computer  
Science, Hanyang Univ.)

※ 본 논문은 한국과학재단 특정기초과제(R01-2003-000-101-50-0)로부터 지원을 받아 진행 하였습니다.  
접수일자: 2006년5월17일, 수정완료일: 2006년9월12일

트 제어기를 살펴보고, III장에서는 Test-Ready PCI 및 Test-Ready USB 인터페이스 제어기의 구조와 동작에 대해 설명한다. IV장에서는 기존의 테스트 제어기와 비교하고, 마지막 장에서 결론을 기술한다.

## II. IEEE 1149.1과 테스트 제어기

### 1. IEEE 1149.1

IEEE 1149.1은 보드 수준에서 테스트 패턴 인가 및 결과 관측을 위한 테스트 표준이다<sup>[1]</sup>. 그림 1은 IEEE 1149.1 경계스캔의 구조를 보여주고 있다. TAP (Test Access Port), 테스트 데이터레지스터, 명령어 레지스터, 그리고 TAP 제어기로 이루어져 있다. TAP은 TCK (Test Clock), TMS (Test Mode Select), TDI (Test Data Input), TDO (Test Data Output), 그리고 선택적으로 TRST\* (Test Reset) 포트로 구성된다. TCK는 테스트를 수행할 때 사용하는 클럭이며, TMS는 TAP 제어기의 상태 천이를 제어하는 신호이다. TDI는 테스트 패턴 인가를 위한 것이며, TDO는 테스트 결과 관측을 위한 것이다. TRST\*는 선택 사항으로서 TAP 제어 상태를 초기화하는데 사용한다. 필수적인 테스트 데이터 레지스터는 칩의 I/O를 위한 경계스캔 레지스터와 TDI로 들어오는 신호를 TDO로 곧바로 패스하기 위한 바이패스 레지스터이며, 선택적으로 디바이스 ID 레지스터와 사용자 정의 테스트 데이터 레지스터를 사용할 수 있다.

그림 2는 TAP 제어기의 16개 상태와 TMS 신호에 따른 상태 천이도를 보여준다. TRST\*가 없을 경우, TCK의 상승에 의해 동기화하여 TMS 신호로 1을 5번 입력하면 초기화 된다.

IEEE 1149.1 동작의 제어는 비트 단위로 TMS 신호를 입력하여 이루어진다. 따라서 보드 테스트는 TMS 신호에 따라 TDI에 테스트 패턴을 직렬로 인가하고 TDO로 나오는 결과를 관측함으로써 이루어진다. 이러한 제어구조에서, 호스트 시스템은 TCK, TMS, TDI 신호를 TAP으로 보내고 다시 TAP으로부터 TDO 신호를 읽어 들여야 한다. 제어하는 사람은 호스트 시스템에 입력 신호를 한 비트의 오차도 없이 정확히 입력해야 하므로, 이는 기계어 수준의 제어라고 할 수 있다. 따라서 이러한 제어의 어려움을 해결하기 위해 호스트 시스템과 TAP 간에 존재하여 명령어 단위의 제어를 할 수 있는 IEEE 1149.1 테스트 제어기가 제안되었다<sup>[2, 3, 4]</sup>.

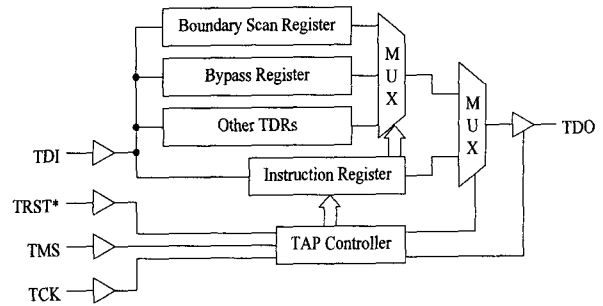


그림 1. IEEE 1149.1 경계스캔 설계 구조  
Fig. 1. Architecture of IEEE 1149.1 Boundary Scan.

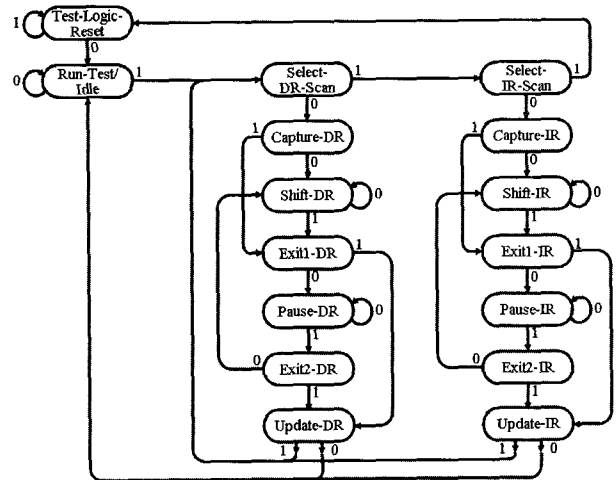


그림 2. TAP 제어기의 상태 천이도  
Fig. 2. TAP controller state diagram.

### 2. IEEE 1149.1 테스트 제어기

TBC (Test Bus Controller)는 TI (Texas Instruments)사의 IEEE 1149.1 테스트 제어기로서, 호스트 시스템과 직접 연결되어 명령어 단위의 제어로 테스트를 수행한다<sup>[2, 3]</sup>.

그림 3은 TBC와 보드의 TAP 사이의 연결 예를 보여준다. 호스트 시스템에서 5비트의 어드레스와 16비트 데이터를 받아 TCK, TMS, TDI 신호를 생성하여 TAP으로 보내고, TAP으로부터 TDO 신호를 받아 호스트 시스템으로 보내는 동작을 통해 테스트를 수행한다.

BSM2(Boundary Scan Master 2)<sup>[4]</sup>는 Lucent 사의

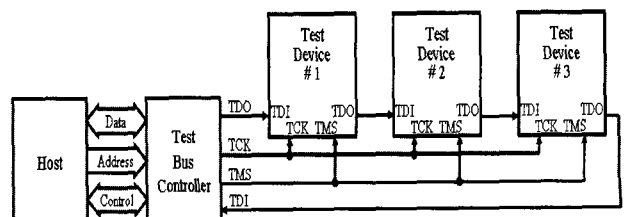


그림 3. TBC와 TAP의 연결 예  
Fig. 3. Example of TBC connection with TAP.

IEEE 1149.1 테스트 제어기로써, 호스트 시스템과 보드의 TAP 사이의 연결 방법은 TBC와 동일하다.

이와 같이 테스트를 명령어 단위로 할 수 있는 여러 가지 IEEE 1149.1 테스트 제어기가 존재하지만 호스트 시스템과 연결하기 위한 방법들에 대해서 고려가 없었다는 문제가 있었다.

### III. Test-Ready PCI/USB 인터페이스 제어기

본 논문에서 제안하는 TRPCI(Test-Ready PCI) 및 TRUSB (Test-Ready USB)는 크게 두 개의 블록으로 나눌 수 있다. 하나는 호스트 시스템과 인터페이스를 위한 각각의 PCI 및 USB 인터페이스 제어 블록이고, 다른 하나는 보드의 TAP과 인터페이스를 위한 IEEE 1149.1 테스트 제어 블록이다.

#### 1. PCI 인터페이스 제어 블록

일반적으로 PCI 버스는 32bits, 33MHz로 동작하며 하나의 호스트 시스템에 여러 PCI 장치가 연결되어 고속의 인터페이스를 구현하고 있다. 따라서 현재 대부분의 서버 및 PC 등의 호스트 시스템에서 PCI 버스를 지원하고 있다.

PCI 인터페이스 제어기는 Target과 Master의 두 가지 기능으로 분류된다. Target은 버스 사용권을 획득할 수 없어 Master에 종속되어 동작한다. 이런 경우에 Master의 역할은 주로 호스트 시스템의 CPU가 담당하게 된다. Master는 Target의 기능을 모두 포함하며, 어드레스, 데이터, 제어신호를 통해 독자적으로 버스를 제어할 수 있다<sup>[6]</sup>.

TRPCI 인터페이스 제어기는 PCI Target 기능으로써,

PCI Spec. 2.2에 호환되도록 설계하였고 호환성에 대한 검증은 PCI SIG에서 제공하는 PCI Compliance Checklist - Revision 2.2에 의한 시나리오를 기초로 하였다<sup>[7]</sup>.

#### 2. USB 인터페이스 제어 블록

대표적인 PC의 시리얼 인터페이스장치인 USB는 패킷 단위로 데이터를 전송하도록 되어 있으며, PCI와 유사하게, PCI의 Master역할을 수행하는 Host와 그에 접속하는 주변장치를 Client의 개념으로 두어 주로 PC가 Host의 역할을 수행할 수 있도록 되어 있다. 데이터의 전송방식은 엔드 포인트라는 특정 타입에 대응되는데, 이러한 특정타입은 Control, Bulk, Isochronous, Interrupt의 네 가지로 구성되며, 서로 다른 데이터의 크기와 용도를 가지고 있다. 또한 엔드 포인트는 입력 또는 출력을 정하는 방향성과 주소 역할을 수행하는 고유의 번호를 가지고 있다.

TRUSB제어기는 PC를 통하여 제어를 받는 Client 구조로 되어 있으며, 데이터의 손실이 없는 Bulk 방식으로 데이터를 주고받게 되어 있다. 이를 구현하기 위하여 Microchip사의 PIC16F877 마이크로프로세서를 이용하여 National Semiconductor사의 USBN9603 USB 트랜시버 칩을 구동하였으며, USB Spec. 1.1에 호환되도록 구성하였다.

#### 3. IEEE 1149.1 테스트 제어 블록

본 논문에서 제안하는 IEEE 1149.1 테스트 제어 블록은 기본적으로 PCI 인터페이스 제어기의 BAR (Base Address Register)중 한 부분의 메모리 영역을 사용하여 동작한다. 또한 PCI 버스를 통해 데이터를 주고받으므로, 32bits의 데이터를 사용한다.

그림 4는 IEEE 1149.1 테스트 제어 블록의 구조를 보여준다. 커맨드 레지스터를 통해 명령어를 입력하며, TDI/TDO 레지스터에 32bits의 TDI/TDO 신호를 저장한다. TAP I/O 제어기는 커맨드 디코더와 현재 TAP 제어기의 상태를 알려주는 TAP 제어 상태기에 의해 TCK, TMS 신호를 생성하고, TDI/TDO 레지스터와 TDI/TDO를 연결시킨다.

또한 명령어 수행이 완료된 경우, 다음 명령을 받아 실행하기 전까지 TCK 신호를 '0'으로 만들어 의도하지 않은 TAP 제어기의 상태 천이를 방지한다.

그림 5는 커맨드 레지스터의 구성을 나타낸다. 8bits의 명령어와 함께 5bits의 클럭 수를 입력하면

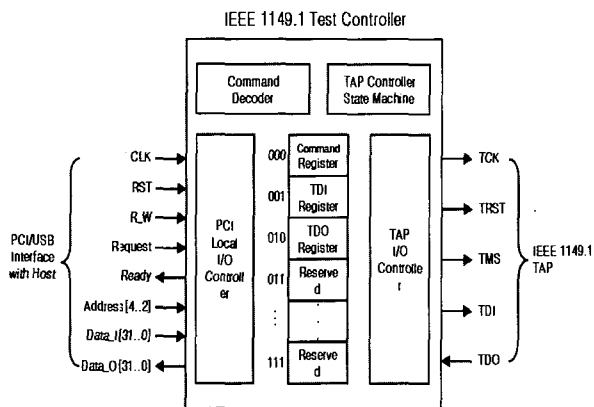


그림 4. IEEE 1149.1 테스트 제어 블록의 구조  
Fig. 4. Architecture of IEEE 1149.1 test controller block.

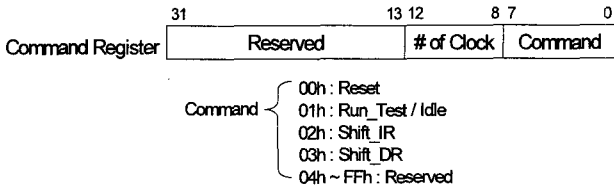


그림 5. 커맨드 레지스터의 구성  
Fig. 5. Consist of command register.

해당 명령을 클럭 수만큼 실행한다. 00h의 Reset의 경우, 입력 클럭 수에 상관없이 5번의 TCK 동안 TMS 신호로 1을 출력하여 TAP 제어기의 상태가 Test-Logic-Reset이 되도록 한다. 이 때, TDO 레지스터의 값도 모두 0으로 초기화 된다. 01h의 Run-Test/Idle의 경우, TAP 제어기의 상태를 Test-Logic-Reset에서 입력 클럭 수만큼 머무르게 한다. 따라서 BIST 또는 테스트 패턴 입력 후 테스트 수행을 위해 사용한다. 02h의 Shift-IR과 03h의 Shift-DR의 경우, TAP 제어기의 상태를 Shift-IR/DR에서 입력 클럭 수만큼 머무르게 한 후, Pause-IR/DR로 이동 시킨다. 이 때, 입력 클럭수가 0인 경우, Shift-IR/DR을 거치지 않고 바로 Pause-IR/DR로 이동시킨다. 따라서 TDI 레지스터의 값을 원하는 비트 수만큼 TDI로 인가하고 싶을 때 사용하거나 TDO로 나오는 결과를 TDO 레지스터에 저장하고 싶을 때 사용한다.

4. TRPCI/USB 인터페이스 제어기의 응용

그림 6은 본 논문에 사용된 TRPCI/USB 인터페이스 제어기의 구조를 보여준다. TRPCI/USB 인터페이스 제어기가 보드 테스트 용도로 쓰이지 않을 때에는, 일반적인 PCI 또는 USB 인터페이스 제어기로써만 동작하게 된다. 다시 말해, IEEE 1149.1 테스트 제어블럭이 일반 동작에 전혀 영향을 미치지 않는다. 또한

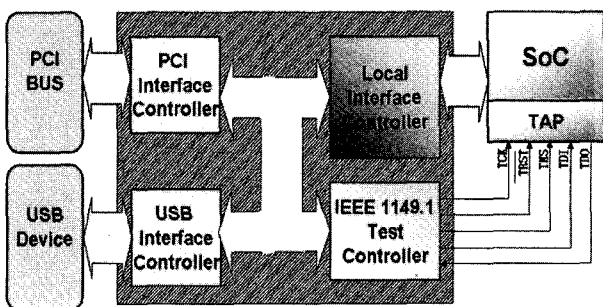


그림 6. TRPCI/USB 인터페이스 제어기의 구조  
Fig. 6. Architecture of TRPCI/USB interface controller.

테스트용으로 사용될 때는 내부에 준비된 스위치를 통해 어떤 인터페이스제어기를 사용해서 테스트를 수행할지 결정할 수 있다. 이에 따르는 세부적인 프로그램을 다음 장에 보인다.

IV. 검증 방법 및 결과

1. Test Ready PCI/USB의 설계

본 논문에서 제안된 IEEE 1149.1 Test Bus Controller 블록은 명령어 단위로 들어온 Data를 Decoding하여 IEEE 1149.1에 필요한 TMS 신호 및 TDI 신호를 적절하게 생성해 내야 함으로 들어온 Data 와 Address를 저장하기 위한 Register와 필요한 신호들을 적절히 형성해 주는 블록, 역으로 TMS를 생성해 주는 블록으로 나누어 설계하였다. 또한 PCI 인터페이스 제어기를 설계하기 위해 PCI 동작을 수행하는 PLX사의 PLX9050 칩을 사용하여 PCI와 기본적인 신호들을 주고받을 수 있도록 하였으며, 각각에 보드에 해당하는 인터페이스 블록을 작성하였다.

또한 USB 인터페이스 제어기의 경우 원하는 USB Spec 1.1에 준하는 동작을 보여주기 위하여, PIC의 마이크로 컨트롤러에 USB를 동작시키기 위한 Firmware를 적재하여, National Semiconductor의 USBN9603 USB Transmitter 칩을 구동하여 USB기능을 수행할 수 있도록 하였다. 또한 32비트 기반으로 설계된 Test Bus Controller에 적합하도록 필요한 신호를 생성해 낼 수 있는 인터페이스 제어기 역시 설계하였다.

기본적인 디지털 로직의 설계는 하드웨어 구현언어로 널리 알려진 Verilog HDL로 설계하였으며, 이의 기능 검증 (Functional Simulation)을 수행하기 위해 Mento사의 ModelSim 버전 5.7에서 검증을 수행하였다.

2. Test Ready PCI/USB의 Function Simulation

TRPCI/USB를 검증하기 위한 전체 SoC 대상코어는 각각 AND 게이트로 구성되어 있으며, 각각 IEEE 1149.1 TAP Controller와 Boundary Scan Cell로 둘러싸여 있다. 이 코어들을 각각 연결하여 TBC와 연결한 회로도도 그림 7과 같다. Board에는 IEEE 1149.1 표준인 TMS, TCK, TRST\_n, TDO 핀으로 구성되어 있으며, TBC에서 생성된 신호들을 통해 테스트가 가능하도록 구성되어 있다. 또한 보드의 I0~I3은 각각의 Core의 입력 핀을, Z는 보드의 최종 출력을 나타내 주고 있다.

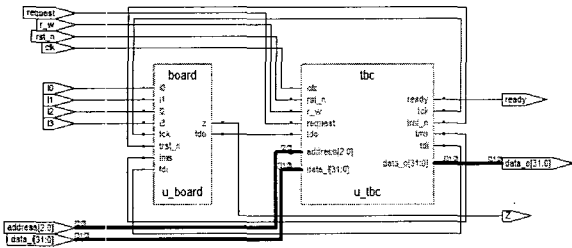


그림 7. TBC와 검증에 사용된 보드와의 연결도

Fig. 7. Connection of TBC and test board.

표 1. INTEST를 위한 테스트 패턴

Table 1. Test pattern for INTEST.

Command	Pattern (16진수)
Reset command	0000 0000
Write TDI Register	0000 0ffd
Shift_IR command	0000 0c02 (12bits shift)
Read TDO Register	0000 0aaa
Write TDI Register	0000 0003
Shift_DR command	0000 0403 (4bits shift)
Run_Test/Idle command	0000 0001
Shift_DR command	0000 0103 (1bits shift)
Read TDO Register	0000 0541
Reset command	0000 0000

표 2. EXTEST를 위한 테스트 패턴

Table 2. Test pattern for EXTEST.

Command	Pattern (16진수)
Reset command	0000 0000
Write TDI Register	0000 0eee (12bits shift)
Shift_IR command	0000 0c02
Read TDO Register	0000 0aaa
Write TDI Register	0000 0009
Shift_DR command	0000 0603 (6bits shift)
Run_Test/Idle command	0000 0001
Shift_DR command	0000 0303 (3bits shift)
Read TDO Register	0000 5403
Reset command	0000 0000

올바른 동작을 확인하기 위하여 IEEE 1149.1의 칩과 칩 간의 연결선을 점검할 수 있는 EXTEST (표2.)와 칩 내부의 동작 상태를 파악할 수 있는 INTEST (표1.)를 수행하였다.

TRPCI/USB 인터페이스 제어기를 사용하여 보드를 테스트하는 순서들은 다음과 같다. ① 커맨드 레지스터에 Reset 명령어를 쓴다. ② TDI 레지스터에 각 칩에 해당하는 IEEE 1149.1 테스트 명령어를 쓴다. ③ 커맨드 레지스터에 Shift-IR 명령어를 클릭 수와 함께 쓴다. ④ 테스트 할 모든 칩에 명령어를 인가할 때까지 ②와 ③의 과정을 반복한다. ⑤ TDI 레지스터에 테스트 패턴

을 쓴다. ⑥ 커맨드 레지스터에 Shift-DR 명령어를 클릭 수와 함께 쓴다. ⑦ 테스트 패턴을 모두 인가할 때까지 ⑤와 ⑥의 과정을 반복한다. ⑧ 커맨드 레지스터에 Run-Test/Idle 명령어를 테스트 수행에 필요한 클릭 수와 함께 쓴다. ⑨ 커맨드 레지스터에 Shift-DR 명령어를 클릭 수와 함께 쓴다. ⑩ TDO 레지스터를 읽어 결과를 관측한다. ⑪ 모든 결과를 관측이 끝날 때까지 ⑨와 ⑩의 과정을 반복한다. ⑫ 테스트 종료 후 일반 동작을 위해 커맨드 레지스터에 Reset 명령어를 쓴다. 다음과 같은 과정으로 SoC설계자는 명령어를 입력하는 ShiftIR 커맨드와 데이터를 입력하는 ShiftDR 커맨드만을 숙지함으로써 복잡한 테스트과정을 간단히 수행할 수 있다는 장점을 가지고 있다.

### 3. TRUSB/PCI통합 검증 환경

그림 8은 TRUSB/PCI의 개략적인 구성도이다. 본 논문에서 사용한 방식을 사용하면, 그림8의 위쪽 테스트 통로의 기존의 UART(RS232C) 포트를 사용하는 Debugging 구조를 대체하여, 보다 빠른 속도를 가지는

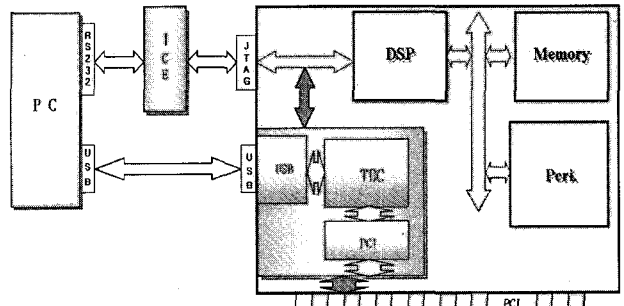


그림 8. 기존의 방식과 제안된 방식의 블록도

Fig. 8. Proposed vs. Existing Method Block Diagram.

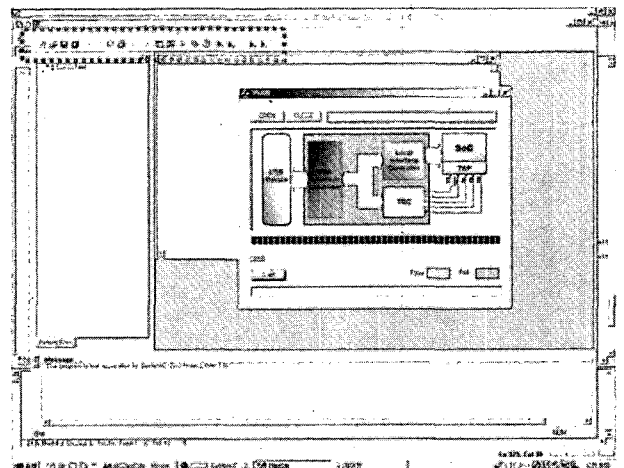


그림 9. IEEE 1149.1 테스트 제어 응용프로그램

Fig. 9. IEEE 1149.1 Test Control application program.

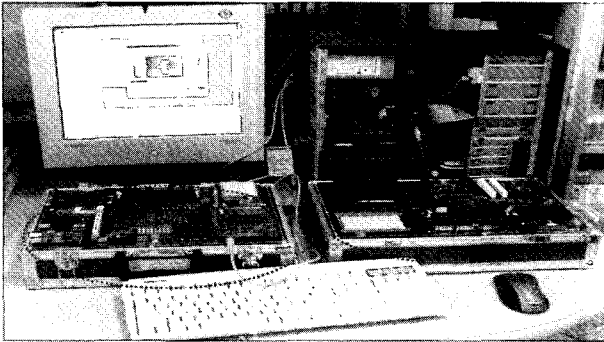


그림 10. TRUSB/PCI 통합검증환경  
Fig. 10. Test Environment of TRUSB/PCI.

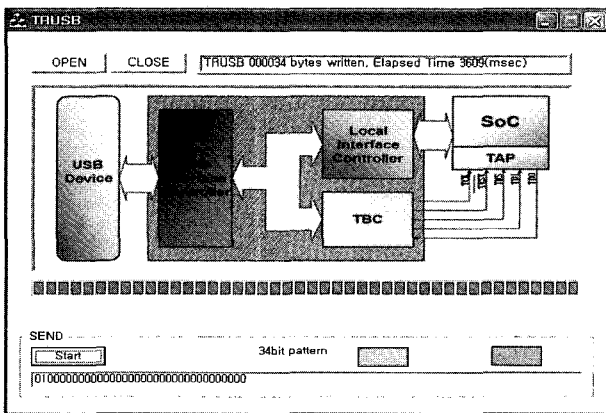


그림 11. TRUSB PC 응용프로그램  
Fig. 11. TRUSB PC application program.

USB/PCI 인터페이스 제어기를 사용하여, Test 및 Debugging을 수행할 수 있다는 장점을 가지고 있다.

따라서 본 논문에서는 다음과 같은 방법으로 실험을 수행하였다. 우선 Test Pattern을 인가하기 위하여 USB와 PCI에 준비된 Pattern을 인가할 수 있는 프로그램을 그림 9과 같이 Visual C++을 기반으로 개발하였다.

그림 9의 프로그램은 PCI 및 USB를 통해 Test Pattern을 인가하기 위한 통합 환경으로서, 지정된 Test Pattern을 불러와서, 자동으로 Test Pattern을 인가하는 기능, 불러온 Test Pattern을 명령어 단위로 한 라인씩 입력하는 기능, 지정된 패턴이 아닌 사용자가 임의의 패턴을 인가할 수 있는 기능 등을 포함하며, 한 프로그램에서 각각의 USB 및 PCI 인터페이스를 선택할 수 있는 기능 등을 포함하고 있다. 이런 일련의 기능들은 그림 9의 점선원 안에 있는 버튼을 클릭함으로써 간단하게 선택되어 질 수 있다.

그림 10은 각각 준비된 Test-Ready PCI 및 Test-Ready USB 인터페이스 제어기보드를 Altera사의 Excalibur ARM을 사용한 검증보드에 각각 물린 모습을 보여주고 있다. 그림 10의 왼편의 원은 USB

Interface제어기를 장착한 검증보드의 모습이며, 오른편의 원은 PCI Interface제어기를 연결한 검증보드의 모습이다.USB Interface 제어기의 예로 간단한 동작에 대해서 설명하면 다음과 같다.

① 파일메뉴에서 지정된 패턴을 프로그램으로 불러온다.

② 패턴이 지정 되면 상태 바 중에 원하는 테스트 옵션을 선택하면 그림 9와 같은 창이 하나 뜨면서 지정된 디바이스로 테스트 패턴을 인가할 수 있는 준비가 된다.

③ Start버튼을 클릭하면 자동적으로 주어진 패턴이 대상 SoC로 인가가 되며, 결과 값이 화면에 표시 된다. 위와 같은 방법을 이용하면, 간단한 방법으로 테스트 패턴을 인가하고 관측할 수 있는 장점이 있다.

그림 11과 같이 대상 SoC에 테스트 패턴을 인가 할 경우 명령어를 통한 테스트 패턴의 인가를 통해 기존의 비트 단위로 데이터를 인가하는 경우 보다 간단한 절차로 동일한 결과를 얻을 수 있다.

결과적으로, TRPCI 인터페이스 제어기를 사용 한다면, 호스트 시스템과 보드의 연결을 쉽게 할 수 있고, 보드 테스트도 간단한 4가지 명령어만을 가지고 쉽게 할 수 있다.

## V. 결 론

기존의 보드상의 연결선 테스트 및 코아의 내부를 테스트하기 위해 제안된 방식들과 본 논문에서 제안한 방식과의 차이점은 표 3과 같다.

표 3과 같이 본 논문에서 제안된 Test Bus Controller는 기존의 TI 방식은 PCI 브리지와 UART칩과 연계하여 사용하도록 되어 있는 반면 본 논문에서

표 3. 기존 방식과의 차이점  
Table 3. Deferencé of existing method.

	TI TBC <sup>[1]</sup>	Lucent BSM <sup>[5]</sup>	Proposed Method
Address/Bus width	5bit / 16bit	lower 16bit / 16bit	32bit / 32bit(3bit)
ASP Support	○	○	○
Number of Registers for TEST	24	18	3
Maximum Freq.	30Mhz	65Mhz	33Mhz/66Mhz
Target Application	UART/PCI	UART/PCI	PCI/USB

제안한 방식은 PCI + TBC 및 USB + TBC 단일 구조로서 각각 1개의 칩으로 구현함으로써, 저가로 테스트 및 검증할 수 있는 환경을 구축할 수 있다.

결과적으로, 본 논문에서 제안된 Test Ready USB/PCI Interface제어기를 사용하면, 보드수준의 테스트에서 필요한 명령어들을 간단히 사용이 가능하게 될 뿐만 아니라, 현재 PC환경에서 가장 많이 사용되고 있는 PCI버스 및 USB환경에서 그림 8과 같은 PC응용 프로그램을 통해서 간단하게 대상 코어에 대해 테스트 패턴을 인가해 볼 수 있음으로, 전문 지식이 없는 설계자도 간단한 방법으로 테스트를 수행할 수 있다. 또한 기존의 방식에 비하여, 하드웨어 영역을 반으로 줄일 수 있는 장점이 있다.

### 참 고 문 헌

[1] TI Test-Bus Controller Application Report -SN74ACT8990, Texas Instrument, August 2000

[2] IEEE Standard 1149.1-2001, "IEEE Standard Test Access Port and Boundary-scan Architecture," IEEE, June 2001.

[3] C. Gibbs, "Backplane test bus applications for IEEE STD 1149.1," In Proceedings, ITC 2003. International Test Conference, pp.167-180, Oct. 2003.

[4] J. Lien and M.A.Breuer "A universal test and maintenance, controller for modules and boards," Industrial Electronics, IEEE Transactions on Volume 36, Issue 2, pp.231-240, May 1989.

[5] F.P. Higgins, R. Srinivasan, "BSM2: Next Generation Boundary-Scan Master," VLSI Test Symposium, Proceedings. 18th IEEE, page 67-72, 2000.

[6] TI Addressable Scan Ports-SN54ACT8996 and SN74ACT8996, Texas Instruments, April 1999.

[7] PCI Local Bus Specification-Revision 2.2, PCI SIG, December 1998.

[8] PCI Compliance Checklist-Revision 2.2, PCI SIG, August 1999.

[9] Baker Art, Lozano and Jerry, "The Windows 2000 Device Driver Book-A Guide for Programmer's," Pearson Education Inc., pp.76-79, 2001.

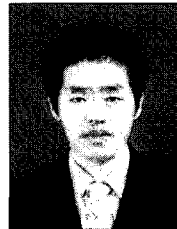
[10] USB Specification-Revision 2.0, USBORG, April 27, 2000.

### 저 자 소 개



김 영 훈(정회원)  
 2004년 한양대학교 전자컴퓨터 공학과 학사 졸업.  
 2006년 한양대학교 컴퓨터공학과 석사 졸업.  
 2006년~현재 삼성전자 System LSI사업부

<주관심분야: ASIC설계, Audio Codec SoC테스팅>



김 기 태(정회원)  
 2005년 한양대학교 전자컴퓨터 공학과 학사 졸업.  
 2006년 한양대학교 컴퓨터공학과 석사 과정 중.  
 <주관심분야: 반도체, 테스트, CAD/VLSI>



박 성 주(정회원)  
 1983년 한양대학교 전자공학과 학사졸업.  
 1988년 University of Massachusetts 전기컴퓨터공학 석사졸업.  
 1992년 University of Massachusetts 전기 및 컴퓨터공학과 박사 졸업.

1992년~1994년 IBM Microelectronics 연구스텝  
 1995년~현재 한양대학교 전자컴퓨터공학부 정교수.

<주관심분야: 테스트 합성, Built-In Self Test, Scan Design, ATPG, ASIC설계, 고속 신호처리 시스템 설계, 그래프이론 등>