New ATE Instrument For PCIe Protocol Testing

A.T. Sivaram

ADVANTEST CORPORATION
Outline

Concepts
Capabilities
Implementation
Use Case
Conclusions
### PCIe Background

- PCI Express is high-performance interconnect protocol defined for a variety of applications on computing and communication platforms. Specifications are developed by PCI-SIG (PCI Special Interest Group).

- PCIe fabric is composed of point-to-point Links that interconnect a set of two components by a dual-simplex communications channel.
  - Root Complex is Root of an I/O hierarchy that connects the CPU/memory subsystem.
  - Endpoint is the Requester or Completer of PCIe transaction.

- PCIe is a layered protocol consisting of three discrete logical layers.
  - Transaction layer
  - Data Link layer
  - Physical layer
FTA Concept

Functional Test Abstraction: Integration of design, simulation and ATE functional test

Design and Simulation Environment

FTA-TBLib
Simulation using tester behavioral model

FTA-VTP
Supports Verilog as common language

FTA-Elink

FTA-TBLib

DUT Model

TestBench

DV Program

Transactor

FTA-TBLib

Common Test Scenario (Verilog)

Initial begin
for (i = 0; i < 4; i=i+1) begin
    Write('h2+i, wdata);
end

for (i = 0; i < 4; i=i+1) begin
    Read('h2+i, rdata);
    if (rdata == wdata) begin
        $display("Test-%d PASS", i+1);
    end
    else begin
        $display("Test-%d FAIL", i+1);
    end
end

end

Packet Sequencer

Procedure

Packet Lib

DUT

T2000 Functional Test Environment

Packet Sequencer
Architected to execute protocol-level patterns

T2000+EPP

T2000 Debug Tools
Strong support for debugging of high-abstract patterns

All Rights Reserved - ADVANTEST CORPORATION 2013/7/17
PCIe Test Using FTA

How verify this test. Address, Payload, Order

# LinkUp
Pcie.linkUp;

# Device Setup
Pcie.execMWrReq
Pcie.execMRdCpl
Pcie.execMWrReq

# Test Execution
Jtag.regio

# LinkDown
Pcie.LinkDown;
FTA-Elink PCIe

EDA

Test scenario

FTA-TBLib
(Tester model)

PCIe Transactor
Pre-Defined Verilog Module

FTA-Procedure

Tester (FTA ATE)

FTA Procedure

PCle Protocol Harness
Pre-Defined Packet Definition for PCIe

DUT

Conversion (Verilog to Procedure)

DUT

Tester model

Pre-Defined Verilog Module

All Rights Reserved - Advantest Corporation

2013/7/17
Connection Interface to Device

• Supports Two Interfaces to connect Device Model
  - Serial Signal (Tx/Rx)
  - PIPE (PHY Interface for the PCI Express)
8GDM: High Level Architecture

- **DDR2**: 256M Vectors, Format/Timing, I/O, 96 SE I/O pins or 48 Diff I/O Pins, 4 segments, up to 2Gbps

- **DDR3**: 256M Vectors, Format/Timing, I/O, 512M Vectors, Format/Timing, I/O, 2Gbps-4Gbps, 64 SE I/O pins or 48 Diff I/O Pins, 4 segments

- **DDR4**: 512M Vectors, Format/Timing, I/O, Serdes Protocol, PCIe Gen2 USB3, 4Gbps-8Gbps, 32 Diff I/O Pins, 4 segments
Main Memory
- Instruction Memory
- Pattern Memory

Vector Generation Control

Packet Sequencer

Sequential Pattern Generator

Pattern Output Control

Frame Processor
- Timing Generator
- Format Control
- Jitter injection
- Clock Data Recovery
- Digital Compare
- Header Hunt

Fail Capture Control

Fail Input Control

Per Pin Capture Memory

Phase data from another channel for Multi strobe

Driver

Comparator

Device Under Test

Central Capture Control

Result Memory

Central Capture Memory

Per Pin Parametric Measurement Unit

4x

96x

Device Power Supply

Pattern List Memory

Memory

Vector Generation Control

Packet Sequencer

Sequential Pattern Generator

Frame Processor

Timing Generator

Fail Capture Control

Fail Input Control

Per Pin Capture Memory

Phase data from another channel for Multi strobe

Driver

Comparator

Device Under Test

Central Capture Control

Result Memory

Central Capture Memory

Per Pin Parametric Measurement Unit

4x

96x
Packet Sequencer for PCIe

• Packet Sequencer
  - Programmable Protocol Generator
  - Supports PCIe layered protocol

• Programing
  - Test Scenario can be described as Application Layer.
  - Transaction, Data Link and Physical operations are implemented in Packet Sequencer.
  - Timings and Levels are specified by OTPL.

• Supports
  - Single Lane for Protocol
  - 2.5Gbps and 5.0Gbps
  - Root complex and Endpoint
  - Transactions:
    - Link Up, Memory, Configuration, I/O, Message
Packet Sequencer:
- 16 packet sequencers in 8GDM
- 4 slices in a module
- 4 Tx/Rx ports per a slice

Pattern Generation
- 250MHz Data Gen, Compare, Capture
- 20bits drive/strobe per a tester period

Receiver
- CDR Training
- Running Disparity

Capture
- Raw 32M Symbols Capturing
- Framed TLP/DLLP/PHY Capturing
- Time Stamp
- Selection(Payload/Framed/All Symbols)

Event for Synchronization
- Analog and DC module
- Another Interface scenario
- State Trigger
Encoding Transmitted TLP Data

Data Buffer

- Main Data Buffer
  - Byte 0:
    - Fmt
    - Type
  - Byte 1:
    - Message Header
  - Byte 7:
    - Message Code

Transaction Packet (TLP)

- STP
- Header
- Data 0
- Data n
- LCRC
- END

Packet sequencer

- Data Buffer
- CRC Generator
- Scrambler
- 8b10b Encoder

DUT

CRC Generator

- Bit order
- 8bits + Control (KD code, Disparity)

Scrambler

- Clock
- Data In
- Data Out
- D0.0
- 000 00000
- 101 01011

8b10b Encoder

- STP(K27.7)
- 111 11011
- (-)110110 1000
- (+)001001 0111
Decoding Received TLP Data

Eye Centering
- Automatic Adjustment

Data Decoder
- Credit Check
- Sequence #
- Packet Detection

Status Control

Data Compare
- Payload
- CRC Generator
- ALU

Data Generator
- Fail Detector
- Data Capture

8B10b Decoder with
- Auto Disparity Sense
- Disparity Error
- Descrambler

DUT
- Alignment
- Payload
- CRC Generator
- ALU

Symbol Alignment
- Expected COM SKP SKP SKP
- Early COM SKP SKP SKP
- Late COM SKP SKP SKP
Debug Tools on ATE

Transaction Sequences and Payload

- Link Up & Training
- Configuration
- Memory Read/Write
- Other Interface

TLP/DLLP/PHY transmitted and received

10bits Symbols transmitted and received
PCIe Implementation Overview

• FTA For PCI-Express Gen2
  - PCI-Express Base Specification Revision 2.1 (2.5Gbps/5.0Gbps)
  - ATE acts as Root Complex and Endpoints.

• Friendly programing model
  - FTA standard library of PCIe.
  - Test designer can focus on the Transaction-Layer operations.

• FTA PCIe library provides transaction functionality
  - Training & Link Up, Configuration
  - Memory Read/Write Request and Completion.
  - I/O Request and Completion
  - Message Request

• TSS GUI tools allow protocol based debug
  - PCIe transaction-sequence scenario.
  - TLP/DLLP frames transmitted and received.
  - Symbols transmitted and received.
## Supported Transactions

<table>
<thead>
<tr>
<th>Transactions</th>
<th>Test Scenario Description</th>
<th>Memo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training &amp; Link Up</td>
<td><code>linkUp; linkDown; linkSleep(...); linkWake(...);</code></td>
<td>Training strobe point by TS1 ordered Set Symbol alignment Link Up to the LO state</td>
</tr>
<tr>
<td>Configuration</td>
<td><code>execCfgWrReq(...); execCfgWrCpl(...); execCfgRdReq(...); execCfgRdCpl(...);</code></td>
<td>Read Request and Completion Write Request and Completion</td>
</tr>
<tr>
<td>Memory Read/Write</td>
<td><code>execMRdReq(...); execMRdCpl(...); execMWrReq(...); execMWrCpl(...);</code></td>
<td>Read Request and Completion Write Request and Completion 32bit / 64bit address</td>
</tr>
<tr>
<td>I/O Read/Write</td>
<td><code>execIORdReq(...); execIORdCpl(...); execIOWrReq(...); execIOWrCpl(...);</code></td>
<td>Read Request and Completion Write Request</td>
</tr>
<tr>
<td>Messages</td>
<td><code>execIRMMsgReq(...); execIRMsgCpl(...); execVMMsgReq(...); execVMsgCpl(...);</code></td>
<td>Message Request Vendor-Defined Messages</td>
</tr>
</tbody>
</table>
Test Scenario With T2000 FTA

- Testing internal logic of DUT
  - Data communication with DUT through I/F
- Testing I/F itself
  - Functionality, Margin, Compliance, …

I/F Test

Data Communication
Data Communication With DUT

**Test Flowchart**

1. **JTAG**
   - Initialize & Setup Target Block
   - Start BIST Engine
   - Read Status

2. **PCI-Express**
   - Upload Test Data to Target Block
   - Download Data from Target Block

**Test Scenario**

- Initialize
- Write Commands
- Send Conditions
- Start self test
- Read Status

**DUT or Model**

- Target Block
- BIST Engine
- Memory
- PCI-Express
- JTAG

**PCI-Express**

- Root Complex
- Test IP
- Test Scenario

**JTAG**

- Test IP
- Test Bench

**PCI-Express End point**

- Upload Data
- Download Data
- Link Training
Version 1.0;
CycPattern Procedure{
    Pxr "PatternLoad_Simple.pxr:DeviceA_PCIE1";
    Timing "samplePCle.tim:Tim_PCIE";
    PinDescription "pindesc_FTA_8GDM_PCIE.pin";
    DeviceA_PCIE1 {
        Main {
            #Wait Device Reset from JTAG
            Wait(jtagIF);
            #LinkUp & Configuration
            linkUp;
            execCfgWr0Req(128, 'b0011, 32'hff_00, _CfgRWDef, _CplDef);
            execCfgWr0Req(128, 'b0011, 32'hff_00, _CfgRWDef, _CplDef);
            execCfgWr0Req(128, 'b0011, 32'hff_00, _CfgRWDef, _CplDef);
            execCfgWr0Req(128, 'b0011, 32'hff_00, _CfgRWDef, _CplDef);
            #Notify to JTAG and Wait TestMode from JTAG
            Notify(jtagIF);
            Wait(jtagIF);

            #Data Read & Write
            execMRdReq32(128, 128'h11111111_22222222_33333333_44444444, _MRWDef, _CplDef);
            execMRdReq32(128, 128'h11111111_22222222_33333333_44444444, _MRWDef, _CplDef);
            execMRdReq32(128, 128'h11111111_22222222_33333333_44444444, _MRWDef, _CplDef);
            execMWrReq32(128, 1024'h11111111_22222222_33333333_44444444, _MRWDef, _CplDef);
        }
    }
}
Conclusion

Concepts
Capabilities
Implementation
Use Case
Conclusions