TABLE OF CONTENTS

TABLE OF CONTENTS ................................................................. 2
MODIFICATION RECORD ............................................................. 3
ACKNOWLEDGMENTS ................................................................. 4
INTRODUCTION ............................................................................. 5
GROUP 1: PCIE INITIALIZATION AND FUNCTIONALITY ............... 7
  TEST #1.1: PCIE INITIALIZATION ...................................................... 8
  TEST #1.2: PCIE REGISTERS .......................................................... 9
  TEST #1.3: PCIE FUNCTIONALITY .................................................. 10
ACKNOWLEDGMENTS

The University of New Hampshire would like to acknowledge the efforts of the following individuals in the development of this test suite:

David Woolf           UNH InterOperability Laboratory
INTRODUCTION

The University of New Hampshire’s InterOperability Laboratory (IOL) is an institution designed to improve the interoperability of standards based products by providing an environment where a product can be tested against other implementations of a standard. This particular suite of tests has been developed to help implementers evaluate the interoperability of their PCIe products, including motherboards, add-in cards, and bridges.

These tests are designed to determine if a PCIe product designed to specifications defined in PCIe Base Specification Revision 3.0 (hereafter referred to as the “PCIe Standard”) with be interoperable with other products designed to the same standard. Successful completion of all tests contained in this suite does not guarantee that the tested device will successfully operate with all other PCIe products. However, when combined with satisfactory operation in the IOL’s PCIe conformance test services, based on the PCI-SIG Compliance Program, these tests provide a reasonable level of confidence that the PCIe product will function properly in many PCIe environments.

The tests contained in this document are organized in order to simplify the identification of information related to a test, and to facilitate in the actual testing process. Tests are separated into groups, primarily in order to reduce setup time in the lab environment, however the different groups typically also tend to focus on specific aspects of device functionality.

The test definitions themselves are intended to provide a high-level description of the motivation, resources, procedures, and methodologies specific to each test. Formally, each test description contains the following sections:

Purpose

The purpose is a brief statement outlining what the test attempts to achieve. The test is written at the functional level.

References

This section specifies all reference material external to the test suite, including the specific subclauses references for the test in question, and any other references that might be helpful in understanding the test methodology and/or test results. External sources are always referenced by a bracketed number (e.g., [1]) when mentioned in the test description. Any other references in the test description that are not indicated in this manner refer to elements within the test suite document itself (e.g., “Appendix 5.A”, or “Table 5.1.1-1”).

Resource Requirements

The requirements section specifies the test hardware and/or software needed to perform the test. This is generally expressed in terms of minimum requirements, however in some cases specific equipment manufacturer/model information may be provided.

Last Modification

This specifies the date of the last modification to this test.

Discussion

The discussion covers the assumptions made in the design or implementation of the test, as well as known limitations. Other items specific to the test are covered here as well.

Test Setup

The setup section describes the initial configuration of the test environment. Small changes in the configuration should not be included here, and are generally covered in the test procedure section (next).

Procedure

The procedure section of the test description contains the systematic instructions for carrying out the test. It provides a cookbook approach to testing, and may be interspersed with observable results.

Observable Results

This section lists the specific observables that can be examined by the tester in order to verify that the SAS target is operating properly. When multiple values for an observable are possible, this section provides a short
discussion on how to interpret them. The determination of a pass or fail outcome for a particular test is generally based on the successful (or unsuccessful) detection of a specific observable.

**Possible Problems**

This section contains a description of known issues with the test procedure, which may affect test results in certain situations. It may also refer the reader to test suite appendices and/or other external sources that may provide more detail regarding these issues.
GROUP 1: PCIe Initialization and Functionality

Overview:
This group of tests verifies the interoperability of a PCIe system.

Scope:
Comments and questions regarding the implementation of these tests are welcome, and may be forwarded to David Woolf, UNH InterOperability Lab (david@iol.unh.edu).
Test #1.1: PCIe Initialization

**Purpose:** To determine that a PCIe Root Complex and PCIe Endpoint or Switch can properly initialize a PCIe link.

**References:** PCIe Specification

**Resource Requirements:**
- PCIe Root Complex
- PCIe Endpoint or Switch

**Last Modification:** September 26, 2013

**Discussion:** Interoperability testing helps determine that all components of a system are properly working together. In addition to PCIe variable such as data rate, number of lanes, number of slots, connector style, and trace length. There are additional variables such as BIOS version, operating system, driver version that play a role in a fully functioning PCIe system.

**Test Setup:** A PCIe Root Complex and Endpoint are connected. This may be accomplished using a traditional CEM connector, a disk drive connector, or any other mechanical interface that supports PCIe.

**Procedure:**
1. The Root Complex and Endpoint are connected within a system. The system is completely powered off.
2. Power on the system. Allow all necessary steps to complete for the system to boot to the supported operating system.
3. Install any necessary drivers. Determine that the operating system detects the endpoint properly.
4. Repeat steps 1-4 for all slots in the system.

**Observable Results:**
- Verify that the operating system identifies the endpoint.

**Possible Problems:** Methods for determining the observable results will vary between operating systems.
Test #1.2: PCIe Registers

**Purpose:** To determine that a PCIe registers contain expected values.

**References:** PCIe Specification

**Resource Requirements:**
- PCIe Root Complex
- PCIe Endpoint or Switch

**Last Modification:** September 26, 2013

**Discussion:** Status of the PCIe Link can be determined by checking the Link Status and Link Capabilities registers.

**Test Setup:** A PCIe Root Complex and Endpoint are connected, the system is powered on. The operating system and all necessary drivers are loaded and operational.

**Procedure:**
1. Use any available utility to determine the contents of the Link Capabilities and Link Status registers.
2. Determine the speed of the link between the Root Complex and the Endpoint (i.e. Gen 1 2.5GT/s, Gen 2 5GT/s, Gen 3 8GT/s) on each lane.
3. Repeat steps 1, 2 for all slots in the system.

**Observable Results:**
- Verify that the PCIe link operates at the highest mutually supported data rate on all connected lanes

**Possible Problems:** Methods for checking the Link Status and Link Capabilities registers will vary across different operating systems. If DUT claims a maximum speed at a lane width other than maximum lane width, use a lane reducer to demonstrate that DUT is able to operate at the maximum claimed speed.
Test #1.3: PCIe Functionality

**Purpose:** To determine that a PCIe endpoint can properly perform its intended function within a system.

**References:** PCIe Specification

**Resource Requirements:**
- PCIe Root Complex
- PCIe Endpoint or Switch

**Last Modification:** September 26, 2013

**Discussion:** Proper initialization of the PCIe link is only the first step in determining PCIe interoperability. The endpoint must be able to properly perform its intended function. Function may be as varied as graphics cards, storage RAID cards, Ethernet adapters with TCP offload engines, custom signal generation and capture cards.

It is not possible to define rigid procedures for all types of PCIe add-in cards. Therefore it is expected that the add-in card manufacturer determine what the necessary functions to be tested will be. For a graphics card this may be pushing data to a monitor. For a storage RAID card it may be reading and writing data to an attached storage device. In many cases this will require a third device, which will be referred to as the Adapted Device. The exact form of the Adapted Device will be dependent on the type of add-in card functionality being tested. This Adapted Device may be a monitor, storage device, network device, or other appropriate device.

Since the purpose of this test suite is PCIe interoperability, the functional verification need only be performed using one Adapted Device. It is not necessary to test many Adapted Devices, as this would be better served by interoperability testing focused on that type of interface (i.e. 10 Gigabit Ethernet, Infiniband, DisplayPort etc…)

Not all endpoints will require an Adapted device to test functionality. A PCIe SSD for instance can be functionality tested on its own within the system.

**Test Setup:** A PCIe Root Complex and Endpoint are connected. This may be accomplished using a traditional CEM connector, a disk drive connector, or any other mechanical interface that supports PCIe. The system is powered on and initialized, the operating system has identified the PCIe endpoint. If necessary an Adapted Device is attached to the PCIe endpoint, using an endpoint dependent medium. Two examples are provided below. Different configurations may be necessary for different mediums.

**Figure 1:** Example Functionality test requiring only the PCIe endpoint, and no Adapted Device.
Figure 2: Example Functionality test requiring the PCIe endpoint and an Adapted Device.

Procedure:
4. Connect the PCIe endpoint to any necessary adapter device, consistent with the function of the endpoint.
5. Follow manufacturer instructions for pushing data to and from the adapter device.
6. Repeat steps 1 and 2 for all slots in the system.

Observable Results:
- Verify that the PCIe Root Complex is able to push data through the PCIe endpoint to the adapter device without error, and at a data rate deemed adequate by the manufacturer.

Possible Problems: Methods for determining the observable results will vary widely among different types of add-in cards and adapter devices. Traffic types and patterns between the endpoint and the adapter device may be determined by the endpoint manufacturer.