64 Lane, 16 Port PCI Express® Switch Performance Report

89PES64H16

Notes

Overview

This document presents performance measurements and benchmarking results for IDT's 89PES64H16 64-lane, 16-port peripheral chip, a member of IDT's PRECISE™ family of PCI Express Switching solutions. The PES64H16 has one upstream port and up to fifteen downstream ports. Ports are nominally 4 lanes wide, but two adjacent 4-lane ports can be merged to create a single 8-lane port. The switch is compliant with PCI Express (PCIe®) base specification revision 1.1.

The test vehicle for the PES64H16 is the evaluation board IDT89EBPES64H16 which hosts the PES64H16. Accompanying the throughput performance metrics are descriptions and methodologies outlining the test setup and procedures.

The nature of tests and the equipment used for these tests varies significantly across the spectrum of tests performed. In the interest of readability and searchability the document is divided into various sections. Each section represents a single test suite that employs a single test setup. A single test suite is capable of high-lighting several features of the switch device under test.

Section I provides some insight into issues that can affect the performance of a PCIe device. This includes overhead resulting from the protocol, as well as the architectural decisions made while implementing the PCIe device.

Section II describes a performance test scenario that demonstrates simultaneous multi-peer wire speed throughput capability of the PES64H16.

Special considerations

The device under test is highly optimized for simultaneous peer to peer traffic between all ports of the switch. This is different from the previously common usage model of a PCI Express switch known as the “fan-out usage model” where majority of the traffic occurs between the upstream switch port connected to the root complex chipset and the downstream switch ports connected to endpoints such as NICs and HBAs. In such a fan-out scenario, the upstream port-width determines the maximum amount of traffic that can flow through the switch and it easy to construct benchmarking scenarios to maximize link utilization using easily available off the shelf hardware (servers and endpoints).

The situation is dramatically different when it comes to benchmarking switches designed for high bandwidth simultaneous peer to peer traffic. A benchmarking set up in this situation will require intelligent endpoints or controllers which can originate and sink traffic worth x4 or x8 port widths. Significant amount of software development is required to design functionally realizable scenarios where simultaneous peer to peer traffic is sustained on an ongoing basis. Such devices and software is generally not available off the shelf. IDT has been pursuing the goal of creating such software for devices available in the market today. This task has not been completed as of the writing of the initial revision of this performance report (October 1, 2007). Therefore, this document describes testing done in somewhat artificial settings that are not likely to reflect any real life application, nor does this testing utilize anything other than IDT’s own switches in proprietary traffic generator modes. It should be noted that in spite of these limitations, the tests do prove that the device under test meets the design goal of sustaining close to wire speed performance on each port while all ports are loaded to the limit.

Revision History

October 1, 2007: Initial version.

SECTION I: PCIe Performance Basics

The PES64H16 primarily serves the purpose of high-performance system interconnectivity within complex systems in need of simultaneous peer-to-peer traffic at wire speed. Simply put, the PES64H16 allows up to 16 intelligent PCIe devices to simultaneously communicate with each other at bandwidths reflective of x4 PCIe port width each. Given that nothing ever comes for free, it is presumed that this functionality has some "cost" associated with it in the form of real estate on the system board, power/heat, design complexity, support circuitry/devices (clocks, hot plug controllers, EEPROMs, power regulators, jumpers, etc.), signal integrity and software development. Throughput and latency (system performance in general) is not always intuitive to predict without a reasonable understanding of the system and switching device architecture, the usage model of the switching device, and some basic understanding of the PCIe protocol itself. In this section, some of these elements are introduced to the users of the PES64H16, specifically those users who are new to PCIe and switching. Advanced users of PCIe and switches may skip the remainder of this section.

What Does Performance Mean?

PCIe switch performance can mean different things to different users. The following is an introduction to some basic terminology.

"Raw bits" refers to the total number of bits that go through the switch in any given period of time, regardless of function, source, or destination. The PES64H16 is designed to handle 2.5 Gigabits per Second of raw throughput in each direction on each of its lanes. This results in (2.5 Gbps) x (2 directions) x 24 (lanes) = 120 Gbps of raw switching capacity.

"Switch throughput" is calculated as the useful bits passing through the switch per second after subtracting the 8b/10b encoding/decoding overhead from the total raw bits. Discussion of the 8b/10b mechanism is beyond the scope of this document. It is sufficient to note that two out of every ten bits passing across a PCIe link do not contribute to any meaningful user data and are, therefore, subtracted from the throughput measurement. It must also be noted that this overhead is a feature of the PCIe protocol itself and is not uniquely associated with a switch device per se. For the PES64H16, the "switch throughput" becomes (120 / 10) x 8 = 96 Gigabits per second.

"Switch utilization" is the "switch throughput" less the overhead associated with the PCIe protocol infrastructure. Examples of this type of overhead traffic are TLPs containing no user data (messages related to interrupts, errors, hot plug, power management, vendor defined messages, etc.) and eight types of DLLPs (Ack/NAK, flow control, etc.). This overhead is variable in nature and can sometimes be fine-tuned to meet system requirements by modifying the switch settings, such as the ratio of ACK/NAKs to total packets, etc. In general, however, expect this overhead to be about 15% of switch throughput for the majority of real life systems. "Switch utilization" brings us one step closer to estimating how much user data goes through the switch in a given period of time, but there is one more overhead to consider.

Every data packet is preceded and followed by a variable number of bytes. These bytes include the frame K-code, sequence number, TLP header, ECRC, and LCRC. Once this "framing" overhead (see Figure 1) is deducted from the "switch utilization" number, the resulting performance metric is called the "switch efficiency".

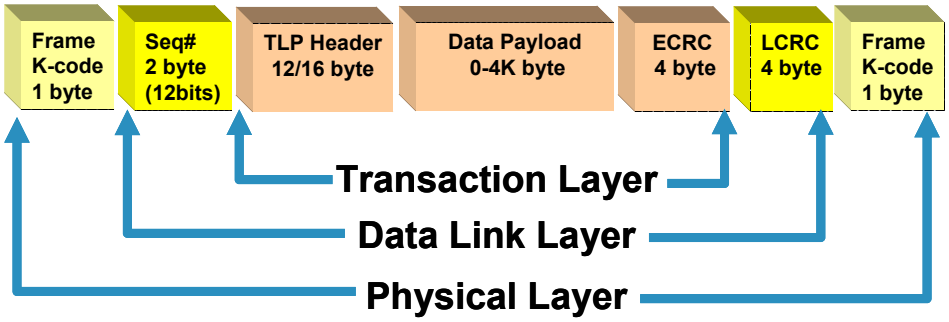

Figure 1 Framing Overhead in a Typical Transaction Packet

A different indicator of the performance of a switch is the switch "latency", which is defined as the time spent by a bit within the switch from the moment it enters the switch to the moment it exits. The latency number, typically low hundreds of nanoseconds, can be affected by several parameters including, but not limited to, switch architecture, traffic pattern, state of the switch in terms of loading, width of the ingress port, and width of the egress port.

Impact of Architecture on Switch Performance

Two high-level architectural decisions which will have the biggest impact on switch performance are "how" the data is forwarded from one port to the other within a switch and "when" the data is forwarded. System designers must make these decisions at the very beginning of the design process. The architectural choices available for the "how to forward" question are: Shared bus, Crossbar, and Shared memory, or a hybrid of some combination of the above. The PES64H16 is implemented in a output queued, shared memory style architecture. Explanation of these different types of switching architectures is beyond the scope of this document.

The architectural choices available for the "when to forward" question are: Cut-through (start forwarding a packet while it is being received) or Store and Forward (start forwarding only after an entire packet is received). The PES64H16 uses the Cut-through forwarding method and can fall back to store and forward mechanism when situations warrant such behavior.

There are several other micro-architectural features or implementation details of a switch that can also have noticeable impact on the performance of a switch. Discussion of the relationship between a feature choice and its impact on performance are beyond the scope of this document. It is relevant to note that several implementation details, such as the transmit retry buffer sizes, ingress buffer sizes, flow control mechanism, allowable maximum payload size (MPS), and controllable frequency of DLLPs including flow control updates and ACK/NACK, have an impact on the performance of the switch. Specifications related to these implementation details for the PES64H16 are found in the 89HPES64H16 User Manual, available by contacting IDT through the helpline at ssdhelp@idt.com.

SECTION II: Simultaneous multi-peer traffic test

The goal of this test is to demonstrate that the PES64H16 (DUT) is able to maintain line rate traffic throughput across all ports under simultaneous multi-peer traffic conditions. The logical flow across all ports is set up in such a way that data entering the switch through one port is switched through the DUT and exits the DUT through three different destination ports in equal measure, as shown in Figure 7. This occurs at maximum link utilization in both directions for each port.

Hardware Setup

As shown in Figure 2, the DUT is populated on a PCB that has 16 PCIe connectors, one per PCIe port of the switch. Each of these connectors is connected to a traffic generator/analyzer. This connection is physically achieved through cables, and at the cables connect to the DUT port connectors through an adaptor card at each end of the cable. The details of this adaptor card are shown in Figure 3 and the connections between ports and traffic generator/analyzer are made as shown in Figure 4.

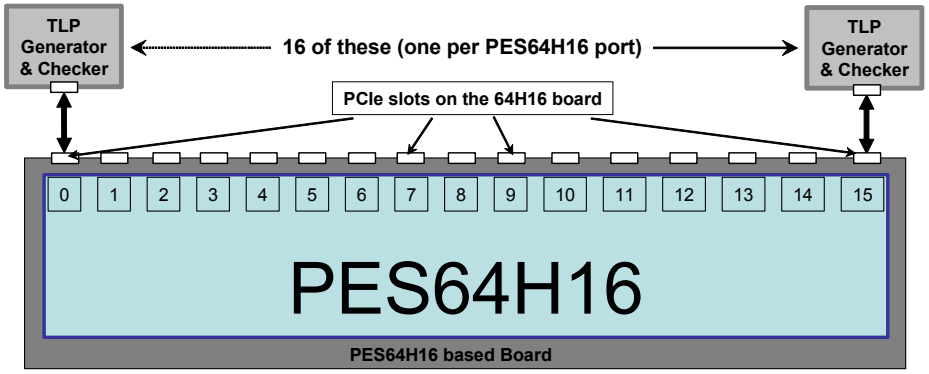

Figure 2 Device under test connected to traffic generator / checker

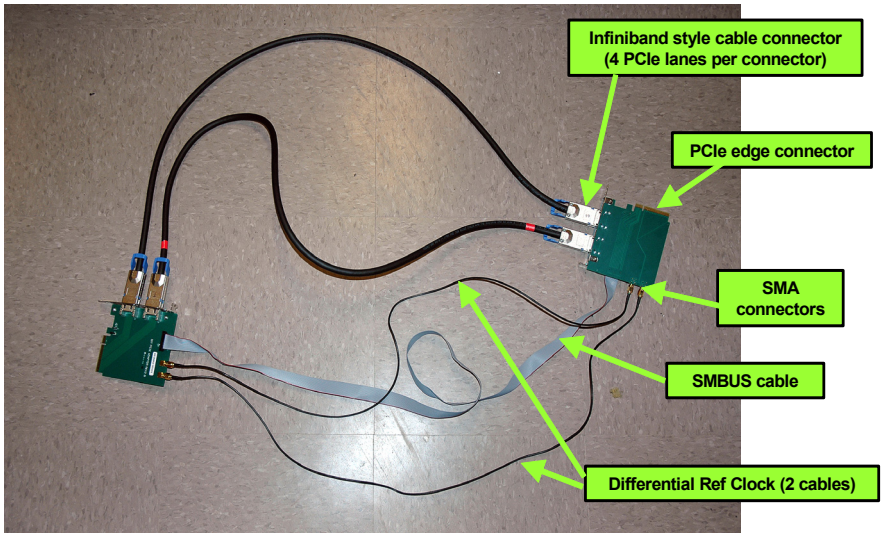

Figure 3 Adapter cards used to connect PCIe slots of different boards

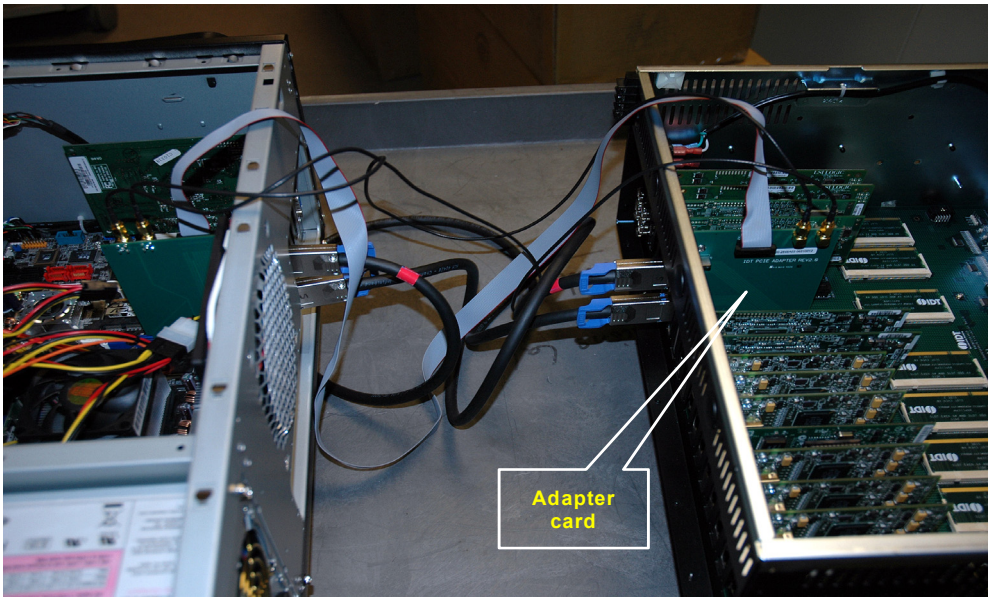

Figure 4 Inserting an adapter card inside a PCIe slot of a board

To accomplish the goals of this test, 16 traffic generators are needed, each one transmitting data to an assigned port of the DUT and maximum link utilization possible. Similarly, 16 traffic checkers are needed to ascertain that data passing through the switch and reaching their intended destination are doing so without any loss along the way. Therefore, what is needed is a single device that can act as traffic generator and checker. This device must be able to generate line rate data in real time and in a manner that is configurable so as to enable the user to define target destinations of various data flows it generates. This device must also enable checking of received data either in real time mode or in post processing mode.

Given that 16 of such devices are needed to enable testing of 16 ports of the DUT, the cost of commercially available PCIe traffic generators/analyzers becomes prohibitive. An easier and flexible solution is required to solve this problem. Proprietary diagnostics features within IDT's own PCIe switch PES32H8 can be deployed as the traffic generator and analyzer. The PES32H8 has a built-in TLP generator that offers limited but sufficient configurability for this test. As shown in Figure 5 and Figure 6, TLPs generated by 3 ports of the PES32H8 are combined into a single stream of three flows to reach maximum link utilization for the link between the PES32H8 traffic generator and the DUT. This stream is transmitted to a single port of the DUT. These TLPS are destined for three different targets after having passed and switched through the DUT. Each such target port of the DUT receives 3 different flows from three different streams/ports, which results in full link utilization for the link between the DUT egress port and the traffic checker connected to it (which is also a PES32H8).

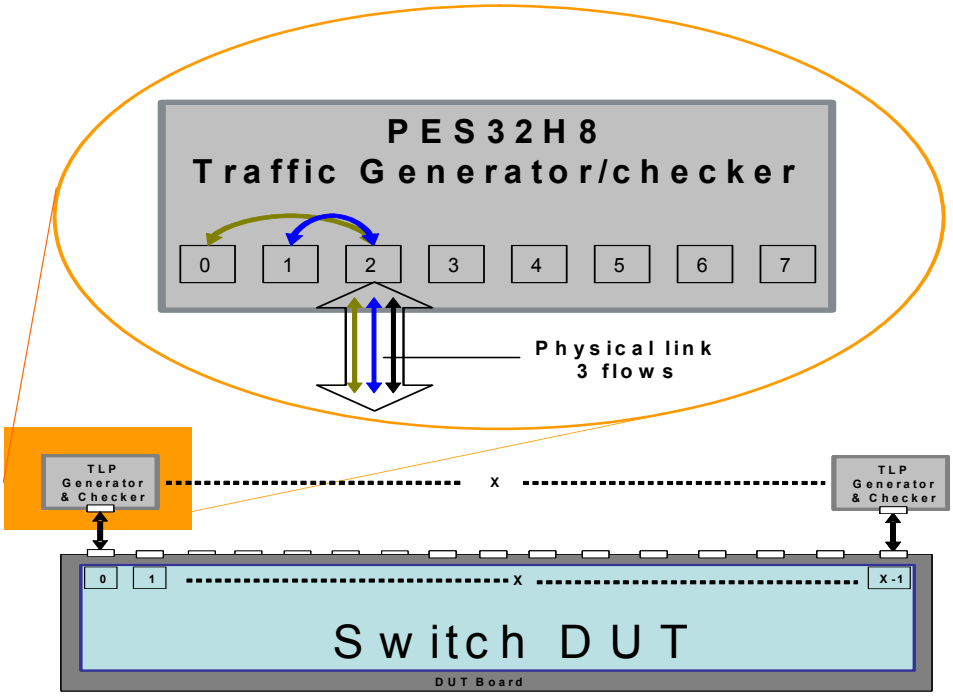

Figure 5 Traffic generator/checker card

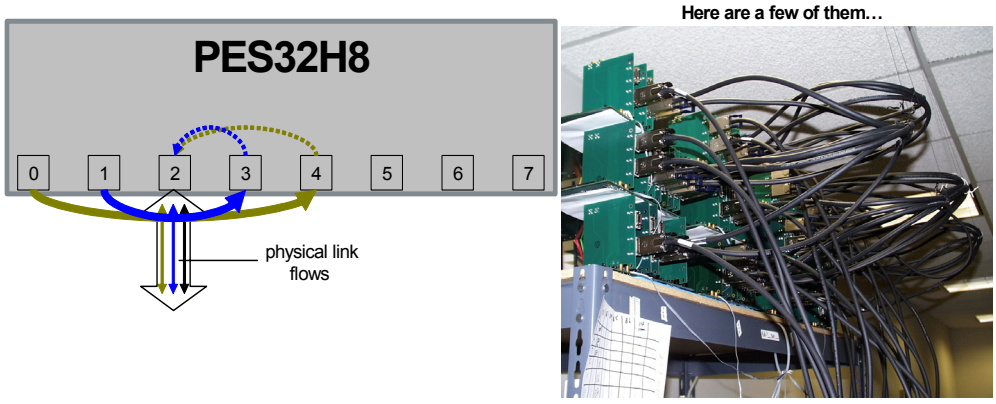

Figure 6 Traffic generator / checker card physical and traffic details

Test Procedure and Methodology

Once the test set up is completed as shown in Figure 8, all traffic generators are powered up along with the DUT. Detailed description of precisely how each traffic generator/checker is set up is beyond the scope of this document. It is sufficient to state that the registers within each 32H8 are initialized in a manner that enables traffic pattern shown in Figure 7. After traffic is enabled from all traffic generators, a steady state of traffic across all ports of the DUT is reached. At this point, link utilization is measured for each ingress and egress port and care is taken to make sure that none of the traffic checkers fail the check. LeCroy PCIe protocol analyzer is used to measure link utilization.

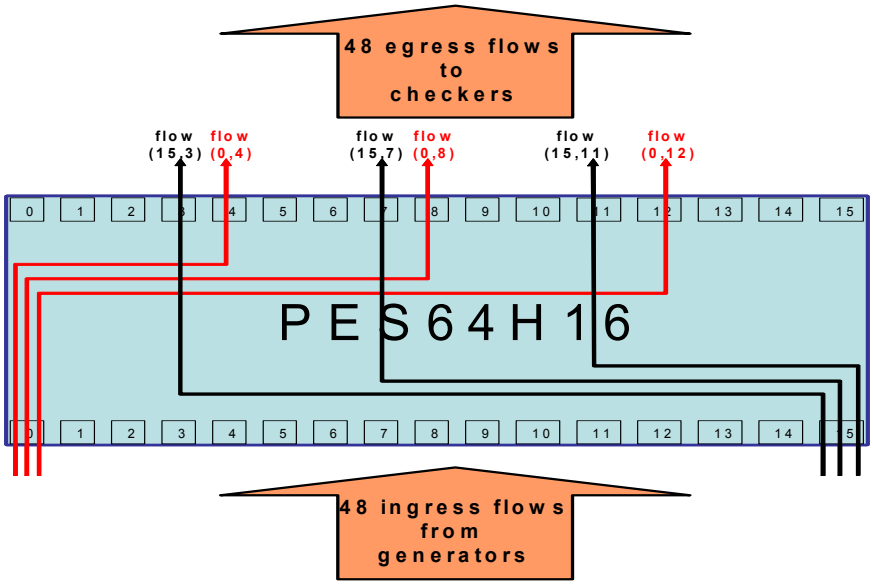

Figure 7 Traffic pattern through the DUT

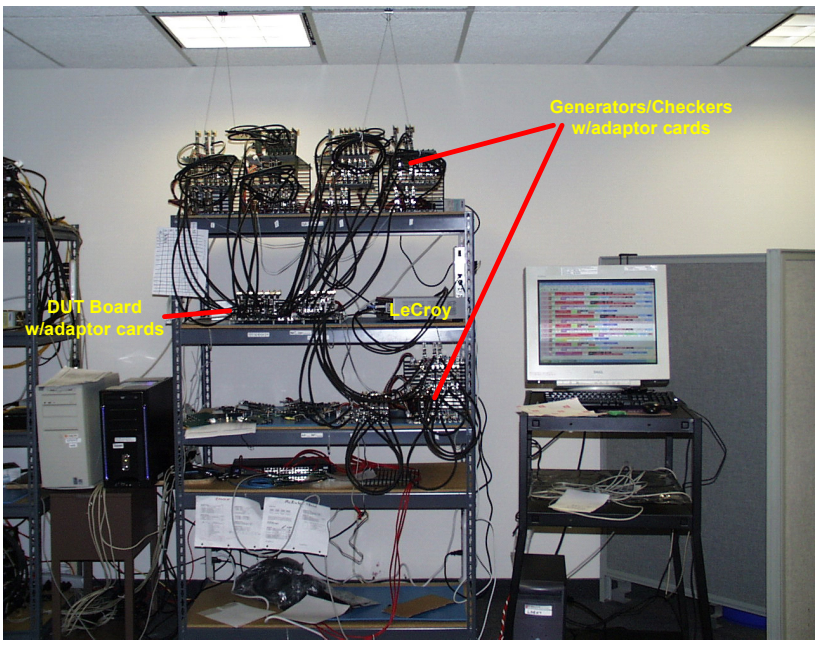

Figure 8 Complete physical set up

Results

Using the LeCroy PCIe protocol analyzer it is seen that all ports are able to achieve close to full link utilization on both ingress and egress side. Port-0 does not have the required connector type to enable LeCroy analyzer to be attached, therefore measurements were done on ports 1 through 15 only. Given the measurements of all other ports it is safe to conclude that Port-0 participated in link utilization to the fullest as well. This baseline for sustained full link utilization as a measure of the switching capability of the DUT is shown graphically in Figure 9.

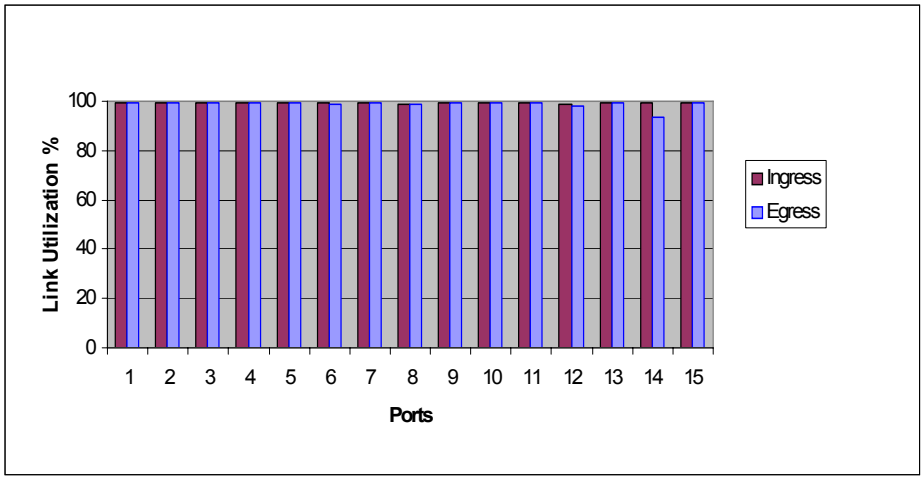

Figure 9 Baseline conditions

A comparison between TLPs measured by the TLP checker per port versus the theoretical maximum possible, is plotted in Figure 10.

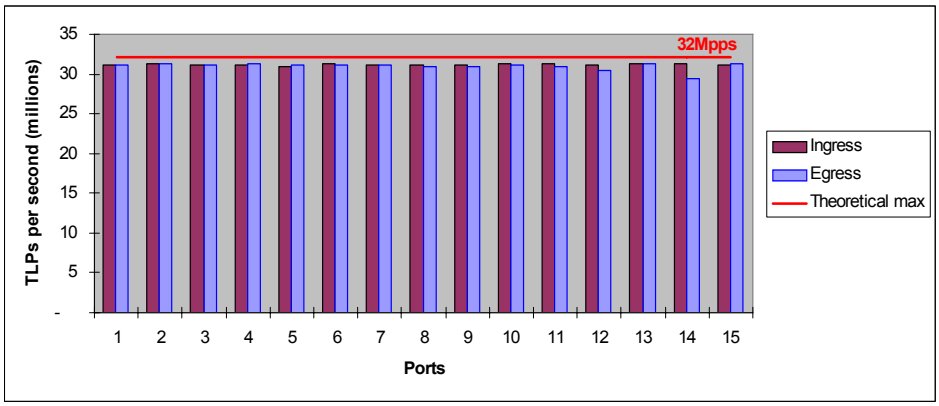

Figure 10 Performance test results

Analysis

For the DUT, the raw wire capacity is 2.5 Gbps in each direction. Removal of 8b/10b encoding overhead results in useful traffic of 2 Gbps. There are 4 lanes per port, which means each port is capable of 8 Gbps (giga-bits per second) or 1 GBps (giga-bytes per second) traffic in each direction. It is safe to assume that two DLLPs of 8 bytes each are sent for every 4 TLPs. This equates to 16 bytes worth of DLLPs per 4 TLPs, or 4 bytes of DLLP overhead per TLP. TLPs are posted and completion packets. Therefore, each TLP is made up of SOF (1 byte), MsgD (20 bytes), Sequence number (2 bytes), LCRC (4 bytes), and EOF (1

bytes), or 28 bytes total per TLP. Consequently, each packet is logically equivalent to 28 bytes of TLLP plus 4 bytes of DLLP overhead, or 32 bytes. Therefore, 1 GBps traffic in each direction translates to 1 GBps divided by 32 bytes, or 32 Mpps (Mega-packets per second). This is indicated by the red line in the plot shown in Figure 10. As can be seen in Figure 10, the DUT is capable of switching close to the theoretically maximum number of packets allowed by the x4 PCIe link.