# 48 Lane, 12 Port PCI Express® Switch Performance Report

*89PES48H12*

## Notes

## Overview

This document presents performance measurements and benchmarking results for IDT's 89HPES48H12 48-lane, 12-port system interconnect PCI Express® (PCIe®) switch, a member of IDT's PRECISE™ family of PCIe switching solutions. The PES48H12 has one upstream port and eleven downstream ports, although the device is specifically designed for highly efficient simultaneous peer-to-peer traffic across all of its ports. Ports are nominally 4 lanes wide (x4), but two adjacent 4-lane ports can be merged to create a single 8-lane (x8) port. The switch is compliant with PCIe base specification revision 1.1.

The test vehicle for the PES48H12 is the evaluation board IDT89EBPES64H16 which hosts the PES48H12. 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 insight into issues that can affect the performance of a PCIe device. This includes over-head 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 PES48H12.

## 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 commonly understood usage model of a PCI Express switch known as the “fan-out usage model” where majority of the traffic occurs between the upstream port of the switch 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 setup in this situation requires intelligent endpoints or control-lers which can originate and sink traffic typically 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 at highest bandwidth. 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 applica-tion, 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 PES48H12 switch 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 PES48H12 allows up to 12 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 PES48H12, 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 introduction to some basic terminology may clarify what ought to be important when selecting a switch for your system design.

### Throughput

"Raw throughput" refers to the total number of bits that pass through the switch in a given period of time, regardless of function, source, or destination. The PES48H12 is designed to handle 2.5 Gigabits per second (Gbps) of raw throughput in each direction on each of its lanes. This results in (2.5 Gbps) x (2 directions) x 48 (lanes) = 240 Gbps of raw switching capacity. In reality, the switch is not required to "switch" this amount of data, as seen below.

PCI express data bytes undergo 8b/10b encoding. 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 stripped off before the data enters the switch core. Therefore, this 20% overhead must be deducted from the raw throughput that the switch must support in terms of actual switching capacity. For the PES48H12, the ideal "switch throughput" now becomes 80% of 240 Gbps, i.e. 192 Gbps, assuming simultaneous bidirectional traffic on all ports.

However, there is more overhead at play. Every payload packet (actual user data) is preceded and followed by a variable number of bytes as required by the PCIe protocol. These bytes include the frame K-code, sequence number, TLP header, optional ECRC, and LCRC. This is the "framing" overhead (see Figure 1).

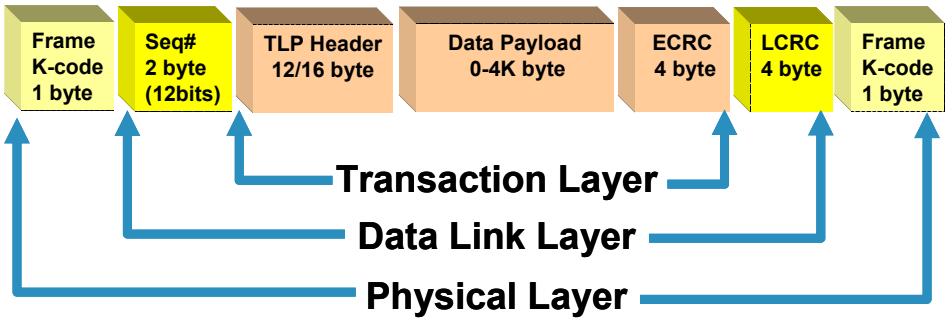


**Figure 1 Framing Overhead in a Typical Transaction Packet**

Figure 2 shows the effect of framing overhead on useful bandwidth for payloads at different PCIe link widths. A 20 byte overhead is assumed per payload packet for the purpose of this chart. This includes 1 byte of start of packet code, 2 bytes of sequence number, 12 bytes of TLP header, 4 bytes of LCRC, and 1 byte of end of packet code.

So, for example, on a x8 link, at 2.5 Gbps per lane per direction, raw bidirectional bandwidth is 40 Gbps. Upon removing the 8b/10b overhead, the useful theoretical maximum bandwidth available is 32 Gbps. Similarly, the theoretical maximum useful bandwidth for a x4 link is 16 Gbps, that for a x2 link is 8 Gbps and for a x1 link it is 4 Gbps.

To understand the calculations behind the chart shown in the figure, let us pick an example of a 64 byte payload packet on a x8 link to see how we come up with the corresponding data point on the chart. Total packet size with overhead becomes 84 bytes on account of the 20 byte overhead explained above. 32 Gbps (giga **bits** per second) of useful bandwidth is the same as 4 GBps (giga **bytes** per second). This is the same as 4000 MBps (mega bytes per second). In terms of packets, this means 4000/84 (i.e. 47.61) million packets. Payload bandwidth for 47.61 million packets is 47.61 multiplied by 64 bytes per packet, or a payload bandwidth of 24.38 Gbps. This is the 64 byte payload data point plotted on the x8 link chart in Figure 2). As seen in the chart, it is possible to achieve close to 32 Gbps (the theoretical maximum for a x8 link) under ideal conditions for payloads larger than 512 bytes.

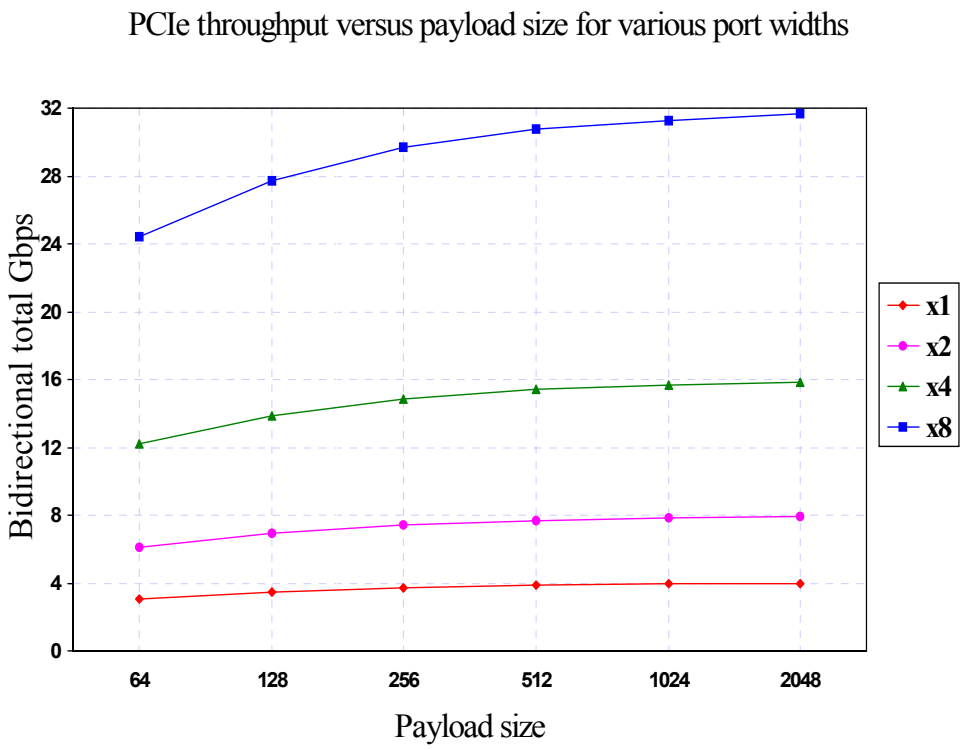


**Figure 2 Effect of framing overhead on link efficiency**

As calculated previously, at 64 byte payloads, the maximum throughput achievable on a x8 link is a bit over 24 Gbps. This means that out of the 32 Gbps useful bandwidth available, approximately 8 Gbps is spent on PCI Express framing overhead and approximately 24 Gbps on payload of 64 bytes payload per packet. This implies close to 75% efficiency on the “wire” (link). Since this framing overhead is constant irrespective of the link width, the wire efficiency is independent of link width in ideal conditions. For those who like to think in terms of wire efficiency as opposed to actual bytes or bits per second bandwidth, Figure 3 can help.

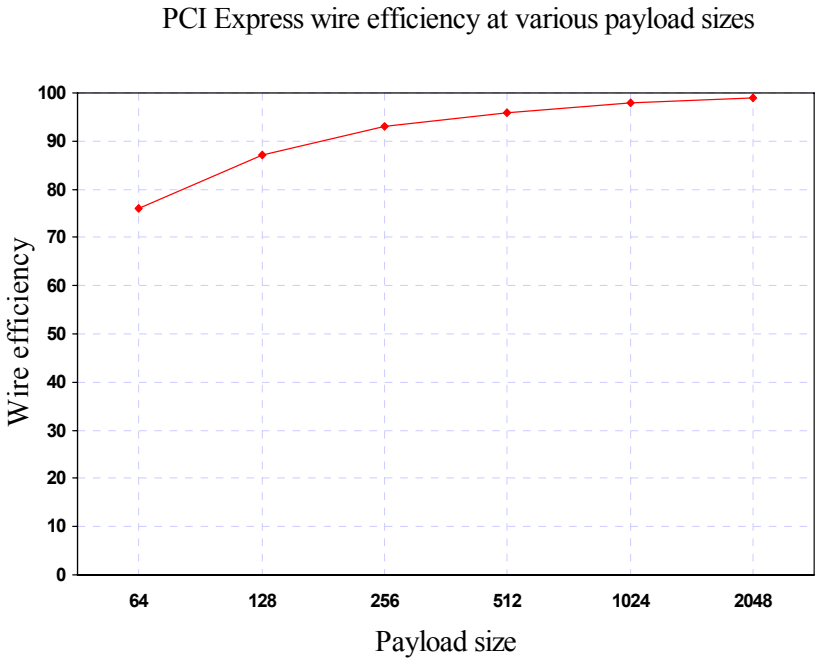


**Figure 3 Data path efficiency of a PCI Express link**

There is more overhead to be considered in addition to the framing overhead. “Switch utilization” is the “switch throughput” described up until this point, 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 or vendor defined messages) 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. Examples of such settings are, the ratio of ACK/NAKs to total packets, frequency of flow control updates, etc. In general, one can expect this overhead to be up to as much as 15% of switch throughput in several real life systems. So, for example, in a x8 link across the switch, for 64 byte payload size, starting from raw bits entering the switch as the base count, 20% is lost in 8b/10 encoding, 25% is lost in framing overhead and approximately 15% may be lost in other protocol overhead as described above.

A pictorial representation of the impact of this additional overhead is shown in Figure 4. This is similar to Figure 3 but also adds another line to the chart showing the effect of the additional DLLPs. The assumption here is that there are two DLLPs of 8 bytes each sent for every 4 TLPs. This equates to 16 bytes worth of DLLPs per 4 TLPs, or on average 4 bytes of DLLP overhead per TLP. This adds to the 20 bytes of framing overhead used previously as an example.

Clearly, the impact of fixed overheads such as these is minimized when the payloads are larger than 256 bytes.

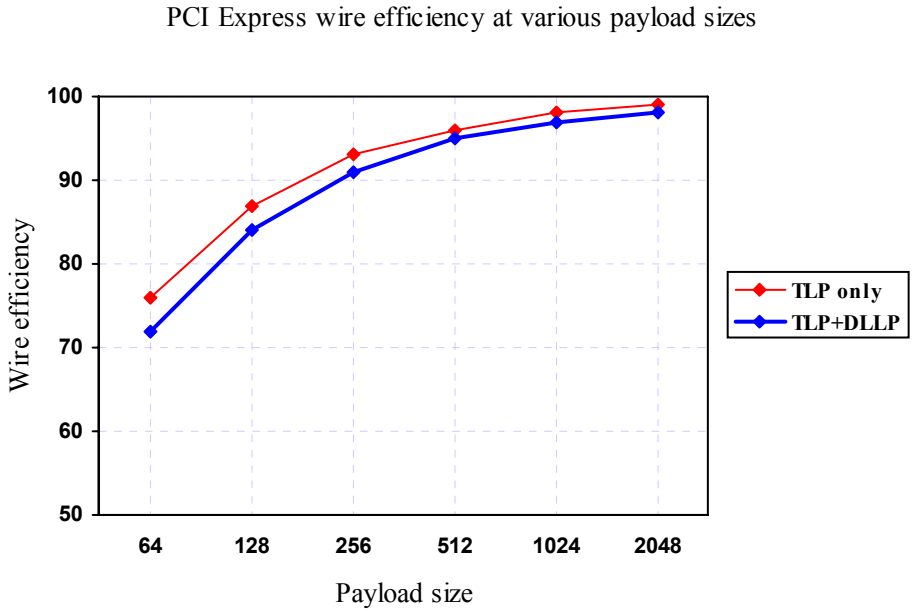


**Figure 4 Effect of DLLP overhead on Data path efficiency of a PCI Express link**

## Latency

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.

It is crucially important to understand what matters and what does not matter when it comes to selecting a PCIe switch on the basis of latency. In general there is little correlation between the latency of a switch and the total throughput it can sustain across all its ports at the same time, which is the metric that truly matters for any system performance. An uninformed chase for a switch with the lowest latency number supplied by a switch vendor can inevitably lead to a wrong decision if no attention is paid to other performance metrics of a switch. Here is why...

**Focus on the port width that matters to the application:**

Some switch vendors tend to mislead customers by providing latency numbers which can only be realized when the switch is configured for the largest port width a switch can offer. In general, wider the port width, lower the latency. For a 48 lane switch, a vendor may offer a low latency number for data passing through the switch from a 16 lane ingress port to a 16 lane egress port. This information is worthless if your application requires data to move form a 4 lane port to another 4 lane port. The key is to focus on latency for the port widths actually required by your application.

**Focus on simultaneous multi-port activity:**

If your application requires data to flow simultaneously between several ports of the switch, what matters is the total latency experienced by the last packet within the set of packets attempting to pass through the switch at the same time. A 12 port switch may have up to 12 different packets trying to get

through the switch at the same instance, one from each port. If a vendor provides the latency for one data transfer across an empty switch, that information is worthless in a scenario such as this. Insist on the total latency for the last bit of the data attempting to go across the switch in a fully loaded condition.

**A well designed system interconnect switch proves its value by offering highest throughput under a condition demanding "simultaneous multiple peer to peer" traffic flows under full load at every port. IDT's PES48H12 switch offers close to theoretical maximum possible throughput (above 95%) under these conditions at all payload sizes. This performance metric is unmatched in the industry as of the writing of this report (October 1, 2007).**

## Impact of Architecture on Switch Performance

Now that the question of what performance means is understood, and what to expect from the PES48H12 is clear, some basic inquiry into how this performance is achieved is in order.

Two high-level architectural decisions which 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, Shared memory, or a hybrid of these. The PES48H12 is implemented in an 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 PES48H12 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 PES48H12 are found in the 89HPES48H12 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 PES48H12 (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 10. This occurs at maximum link utilization in both directions for each port.

## Hardware Setup

As shown in Figure 5, the DUT is populated on a PCB that has 12 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 6 and the connections between ports and traffic generator/analyzer are made as shown in Figure 7.

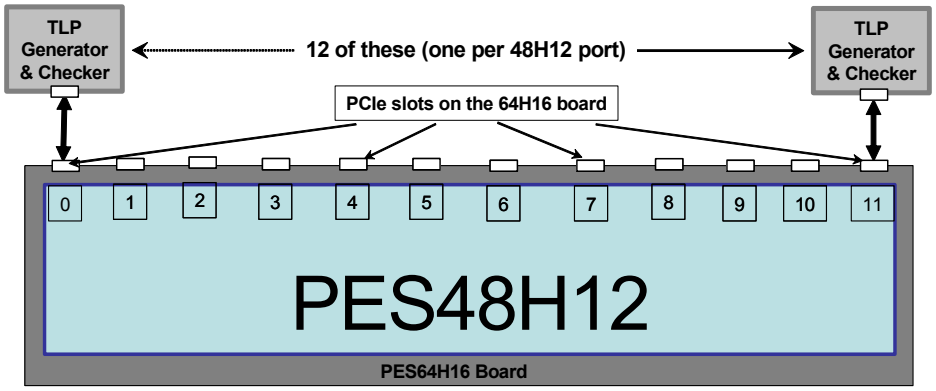


**Figure 5 Device under test connected to traffic generator / checker**

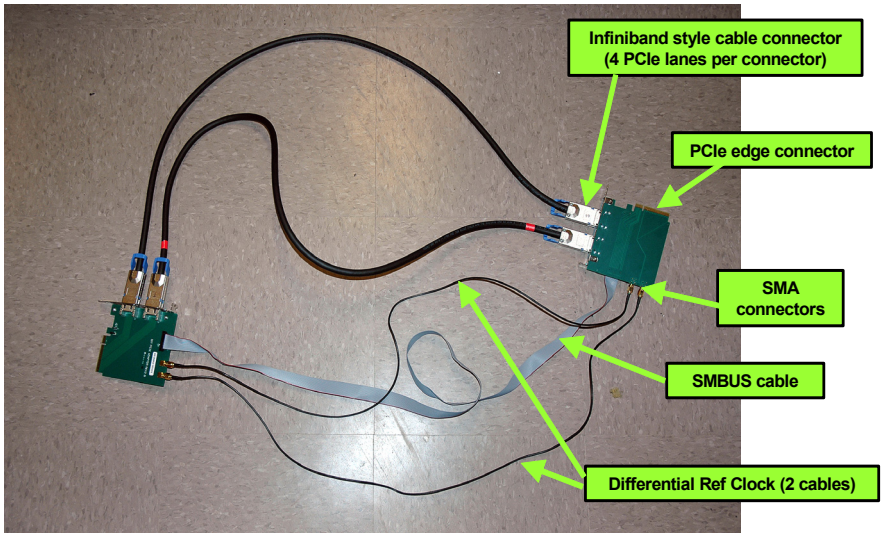


**Figure 6 Adapter cards used to connect PCIe slots of different boards**

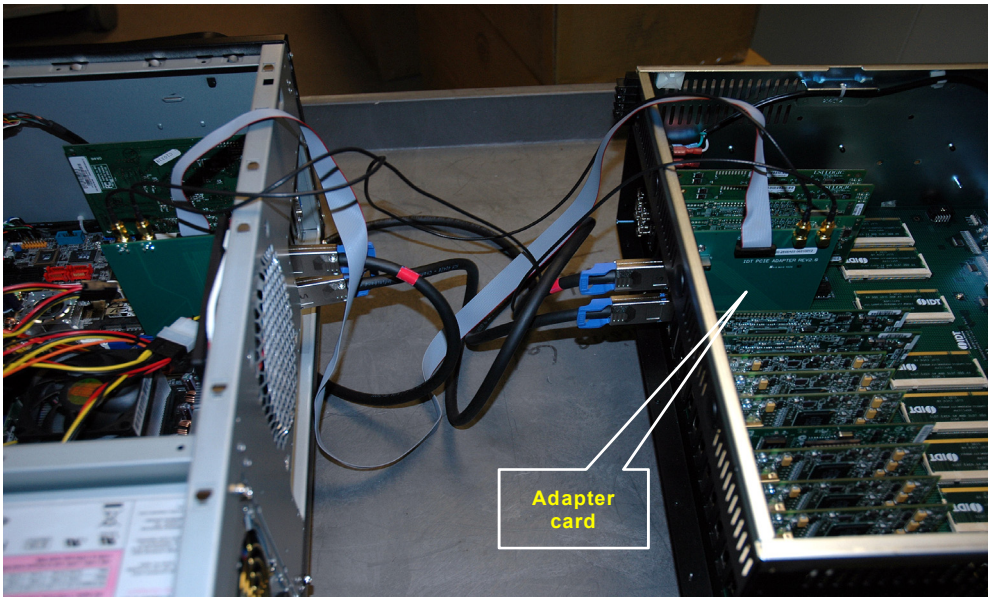


**Figure 7 Inserting an adapter card inside a PCIe slot of a board**

To accomplish the goals of this test, 12 traffic generators are needed, each one transmitting data to an assigned port of the DUT and maximum link utilization possible. Similarly, 12 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 12 of such devices are needed to enable testing of 12 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 8 and Figure 9, 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).

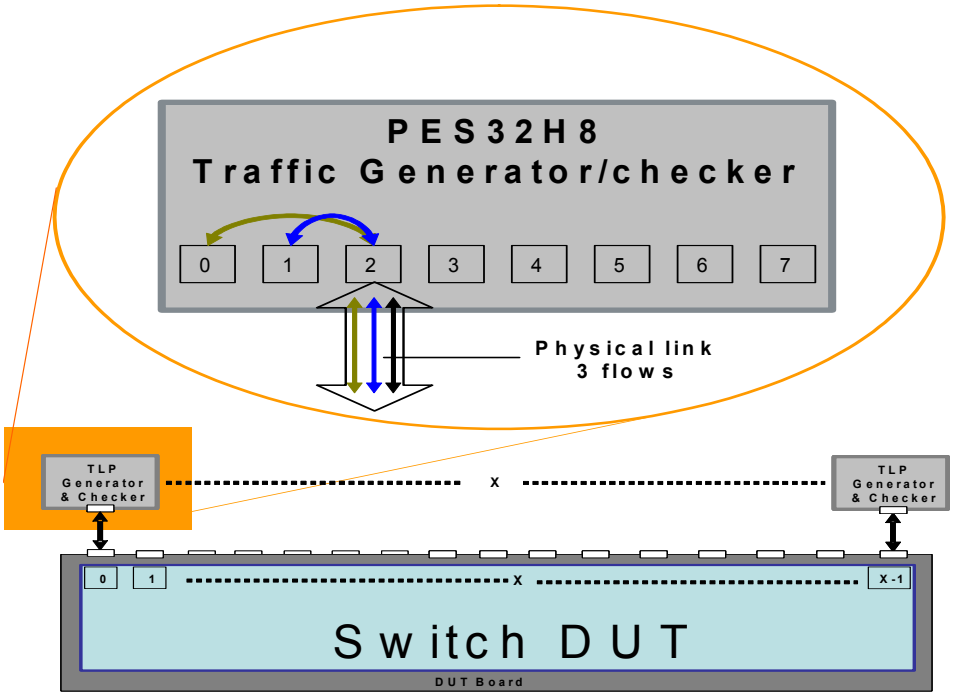


Figure 8 Traffic generator/checker card

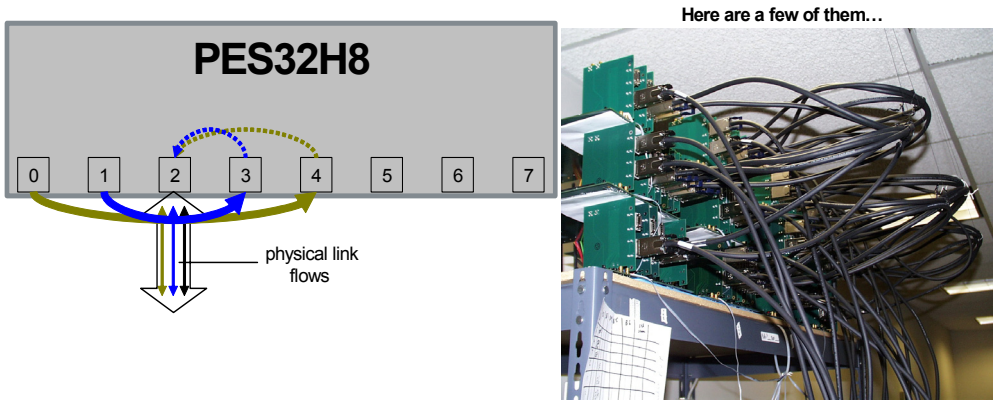


Figure 9 Traffic generator / checker card physical and traffic details

## Test Procedure and Methodology

Once the test set up is completed as shown in Figure 11, 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 10. 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.

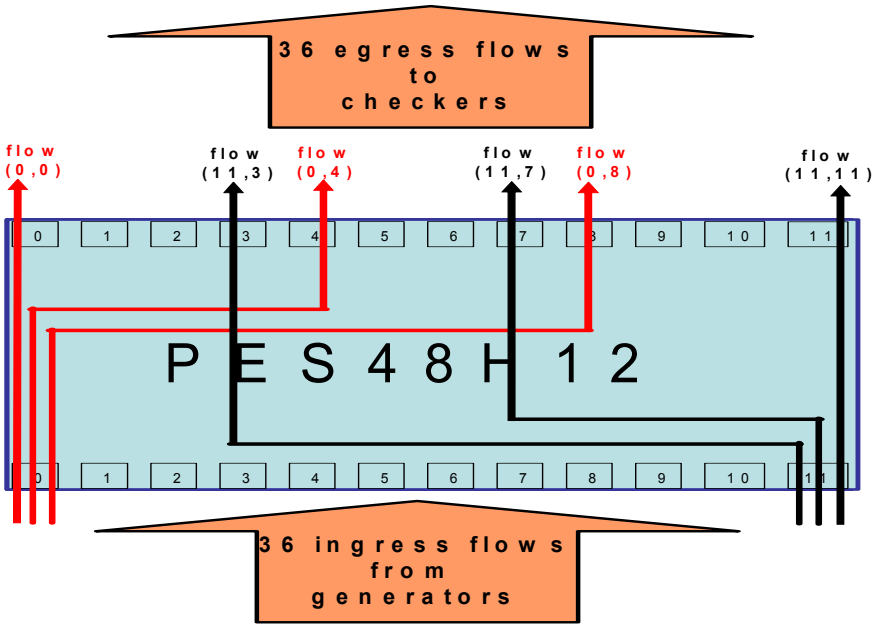


Figure 10 Traffic pattern through the DUT

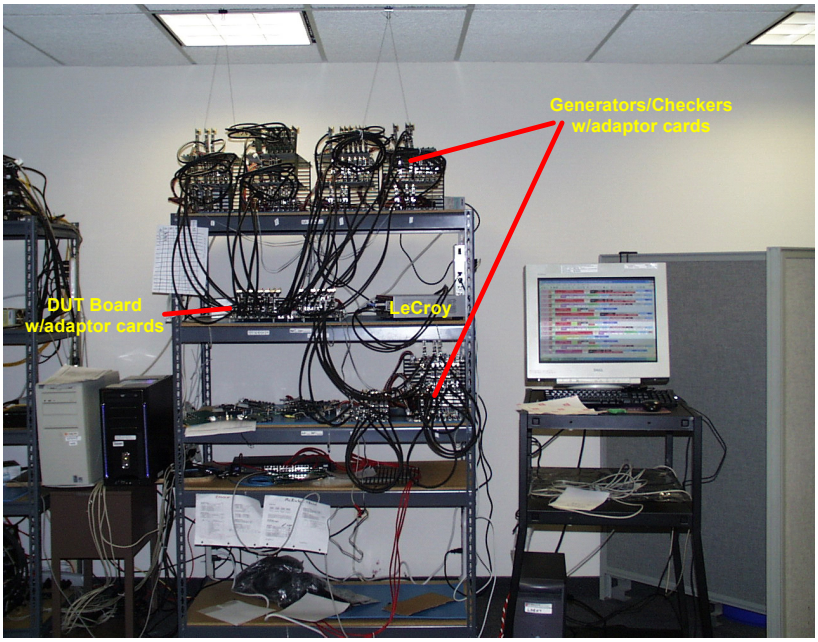


Figure 11 Complete physical set up

## 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 111 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 12.

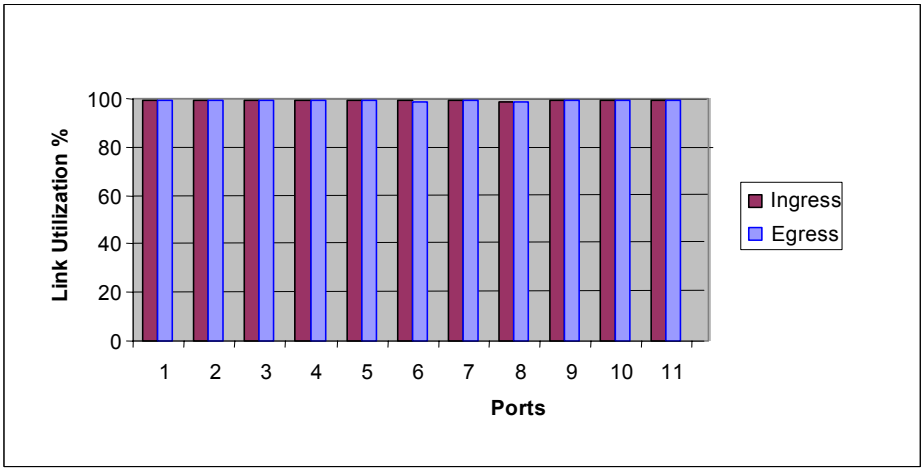


**Figure 12 Baseline conditions**

A comparison between TLPs measured by the TLP checker per port versus the theoretical maximum possible, is plotted in Figure 13.

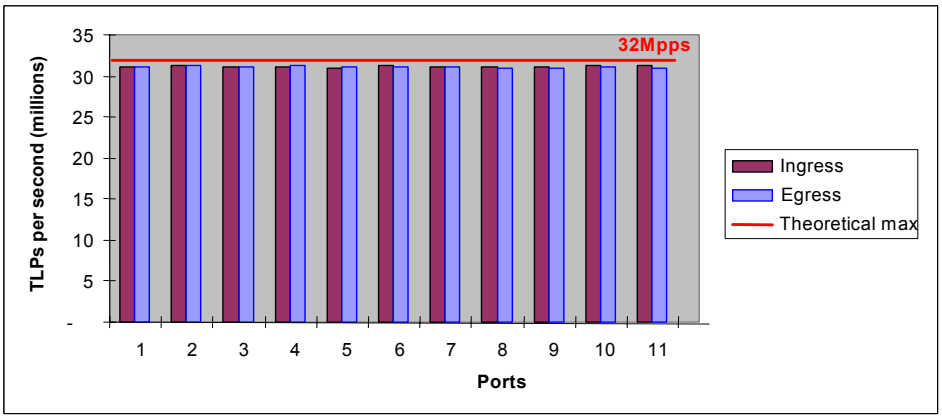


**Figure 13 Performance test results**

## 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 13. As can be seen in Figure 13, the DUT is capable of switching close to the theoretically maximum number of packets allowed by the x4 PCIe link. Given the small payload size used in this experiment, it is safe to assume that any larger payload scenario will lead to similar or better performance, as explained in section I of this document.