White Paper

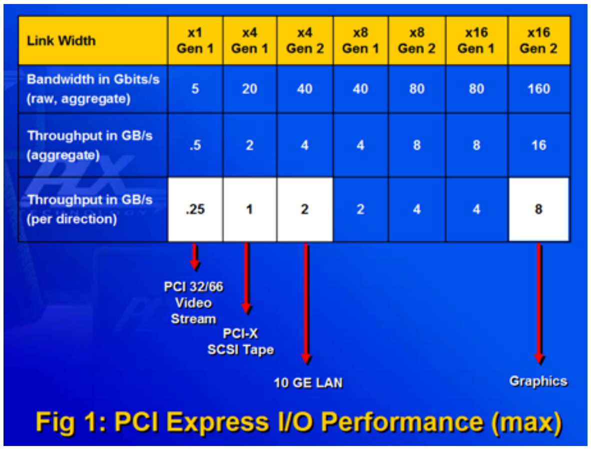
PLX PCI Express Gen 2 Switches Push Performance Boundaries to the Limit

Steve Moore, Sr. PMM, PLX Technology

Today's server, workstation and embedded computing systems are increasingly overwhelmed by the data streaming to and from the various I/O sources; traffic from endpoints like 10 Gig Ethernet, 4X DDR InfiniBand and 8G Fibre Channel continues to drive up the throughput requirement. Graphics cards seem to have an insatiable appetite for speed. Supercomputer and storage clusters require ever-increasing fabric bandwidth. The latest generation of PCI Express (PCIe) switches from PLX is taking on these challenges in a big way, deploying Gen 2 PCI Express speed in new architectures and features such as Read Pacing™, Dual Cast™, and Dynamic Buffer Allocation functions to ease the performance burden on the root complex. This paper looks at how these new features have created dramatic improvements in I/O performance, and offers specific lab and simulation results to help guide the system designer to an optimum result.

The Move to Gen 2

Figure 1 shows how the performance for I/O has scaled from 32-bit PCI through PCI-X and into PCIe. The Gen 2 switches from PLX offer significant increases in available throughput compared with PCI and Gen 1 PCIe devices. Based on the I/O standard maximums, I/O bandwidth capability is multiplied by a factor of 64 times, comparing a 32-bit PCI bus to a x16 PCIe link. In addition to the doubling of line rates (Gen 2 = 5GT/s vs Gen 1 = 2.5GT/s), PLX deployed its fourth-wave of PCIe switch architecture with the introduction of its PEX 8600 series of Gen 2 switches. This architecture allows larger maximum payload sizes (up to 2K) and implements numerous design features which prevent congestive scenarios. The end result is a unique Gen 2 switch product line (the PEX 8600 series) which achieves >99% of the theoretical throughput of a PCIe Express system. In parallel, PLX introduced an inventive software toolbox with extensive monitoring and debug aspects that provide meaningful "fast-to-market" techniques.



Link Width	x1 Gen 1	x4 Gen 1	x4 Gen 2	x8 Gen 1	x8 Gen 2	x16 Gen 1	x16 Gen 2
Bandwidth in Gbits/s (raw, aggregate)	5	20	40	40	80	80	160
Throughput in GB/s (aggregate)	.5	2	4	4	8	8	16
Throughput in GB/s (per direction)	.25	1	2	2	4	4	8

PCI 32/66 Video Stream
PCI-X SCSI Tape
10 GE LAN
Graphics

Fig 1: PCI Express I/O Performance (max)

The *performance*PAK™: Read Pacing, Dual Cast and Dynamic Buffer Allocation for Unmatched I/O Throughput

It can be difficult for today's root complexes to absorb high-speed traffic such as 10 Gig Ethernet, especially when it competes with very fast streaming data from sources such as InfiniBand and Fibre Channel (FC) storage elements. When a few bytes of Ethernet data (which is very sensitive to latency in typical TCP/IP implementations) get stuck behind large packets of FC data in the root complex, the latency that is introduced by this congestion will severely impact system response time and create bandwidth limitations. PCIe Gen 2 switches from PLX have features to mitigate the effects of having to process competing data protocols, thereby improving overall system performance. Read Pacing is one of these features that provides dramatic improvements in system performance in servers and storage controllers.

System Performance Limited by "Endpoint Starvation"

When two or more endpoints are connected to a root complex through a PCIe switch, with unbalanced upstream versus downstream link-widths (and hence unbalanced bandwidths) and an uneven number of read requests are being made by the endpoints, one endpoint inevitably dominates the bandwidth of the root complex queue. The other endpoints suffer reduced performance as a result. This is known as "endpoint starvation," which can make it appear as if the system is congested and not performing optimally.

Figure 2 shows a typical root complex connected to two endpoints through a PCIe switch. In this example, there is a x8 upstream port and two x4 downstream ports. The FC HBA is a good example of an endpoint that could dominate the bandwidth of the root complex queues. In this example, the FC HBA makes several 2KB read requests, which are then queued by the root complex, filling up the queues in root complex. While the queues are full, the Ethernet NIC makes two 1KB read requests. The Ethernet NIC must wait for the root complex to service all of the read requests from the FC HBA before they're serviced. Thus the NIC is "starved."

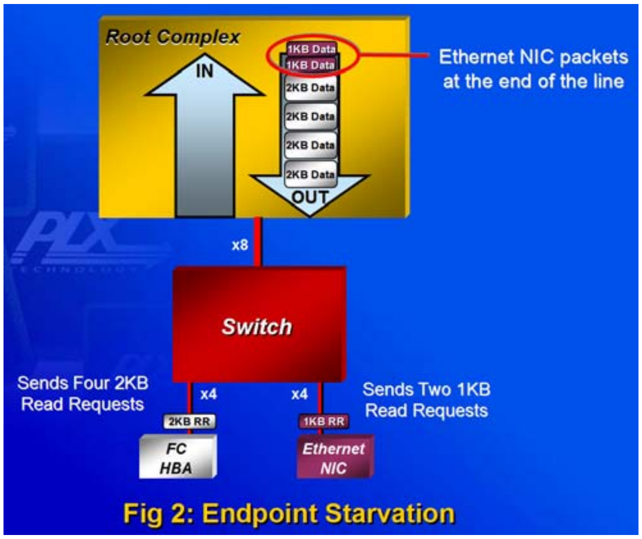


Fig 2: Endpoint Starvation

Read Pacing Feeds the Starving Endpoint

Endpoint starvation is solved – and the endpoint is "fed" -- with a new PCIe switch feature called Read Pacing, which is available on the latest Gen 2 PCIe switches. Read Pacing provides increased system performance with a more balanced allocation of bandwidth to the downstream ports of the switch. With Read Pacing, the switch can apply rules to prevent one port from overwhelming the completion bandwidth or buffering in the system.

With Read Pacing (Fig. 3), the switch controls the number of the FC HBA's read requests forwarded through at a time. Read Pacing allows the switch to balance the outstanding number of read requests with the endpoints ability to process them. Programmable registers in the switch control the number of read requests forwarded to the root complex. As the Ethernet NIC makes its two 1KB read requests, the switch allows both read requests through, thus balancing the flow of data from both endpoints. As shown in Figure 3, a 2KB read for the FC HBA through the root complex is immediately followed by two 1KB reads for the Ethernet NIC, resulting in balanced traffic for each endpoint. Read Pacing allows the Ethernet NIC to be serviced more frequently without impacting the bandwidth of the FC HBA. Hence, endpoint starvation is eliminated with Read Pacing. The chart below compares the performance improvement that can be achieved with and without using Read Pacing in a real world system, where the FC issues 16 4K read requests ahead of the Ethernet single 1K read request.

The PLX 8600 switches have built in defaults for Read Pacing based on the relative link width and speeds of the downstream and upstream ports. Initially, the switch allows reads from the paced port to go at twice the normal read-paced rate to accumulate a number of transactions in the egress port, thus ensuring that the port continues to be fed with traffic at its maximum rate. The Read Pacing keeps sending read requests upstream and keeps ahead of the demand from the paced port.

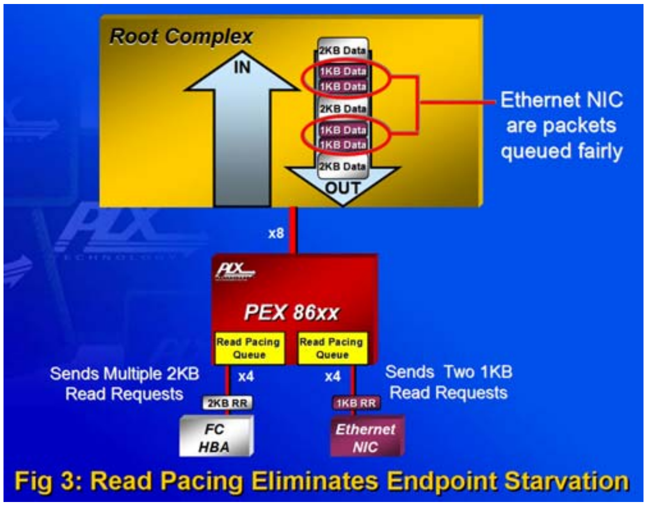


Fig 3: Read Pacing Eliminates Endpoint Starvation

Table 1. Performance Improvement Results with Read Pacing

	Ethernet Throughput	Ethernet Read Latency	FC Throughput
Multi-channel Ethernet only	747MB/s	340ns	-
FC only	-	-	752MB/s
Multi-Channel Ethernet + FC **without** Read Pacing	43MB/s	6200ns	753MB/s
Multi-Channel Ethernet + FC **with** Read Pacing	496MB/s	424ns	753MB/s

Increasing Performance by Optimizing Buffer Size Dynamically

Early PCIe switch architectures provided each port with a fixed amount of buffer RAM. Figure 4 compares a typical type of buffer allocation, seen in the older switch designs, with the new Dynamic Allocation scheme found in the latest Gen 2 switches. In this example, a six-port switch is designed with a total of 30 packet buffers, with five buffer segments available on each port. If only four ports are used, then the buffers allocated to the two unused ports are wasted. Since a larger buffer will translate into better performance, it would be nice if that unused memory could be used to increase the size of the buffers on the four ports that are being used. In the latest Gen 2 PLX switches, it is possible to do just that. This feature is known as Dynamic Buffer Allocation, where a shared memory pool is available to any port, and the size of the buffer is allocated dynamically depending on the number of ports in use.

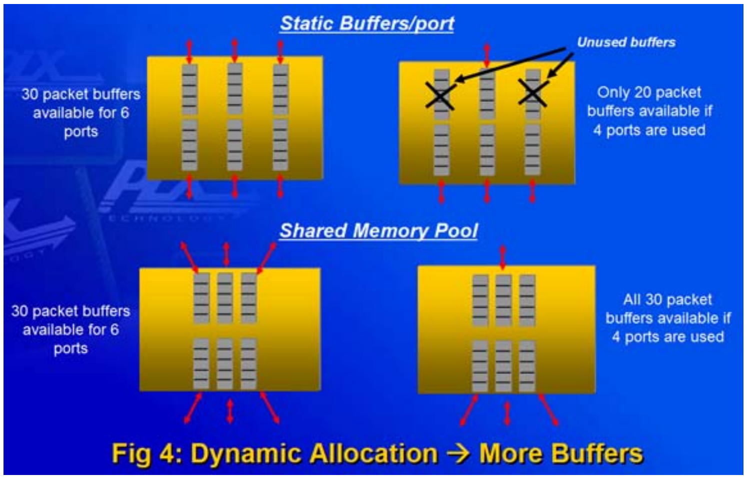

Static Buffers/port
Unused buffers
30 packet buffers available for 6 ports
Only 20 packet buffers available if 4 ports are used
Shared Memory Pool
30 packet buffers available for 6 ports
All 30 packet buffers available if 4 ports are used

Fig 4: Dynamic Allocation → More Buffers

Increasing Performance by Sizing Buffers Dynamically

Figure 5 compares a static buffer per port scheme with a Dynamic Scheme on a switch which is configured with three differing port widths. Since the smaller width ports require less bandwidth than the wider ports, they should require fewer packet buffers as well. In this example, a x8 upstream port is servicing three downstream ports, one a single x1 port, one a x4 port and third one a x8 port. With a static fixed buffer per port architecture, the x1 port is allowed the same buffer size as the x8 ports. Not only is this not the optimal buffer assignment, but there are two unused groups of packet buffers. With Dynamic Allocation, buffers are assigned as needed to each port based on the width of each port. Since there are no unused buffers, a larger total amount of buffer is available, increasing the size of buffer that may be applied in the ports that need the extra bandwidth. In this example, in the bottom half of Figure 5, ten packet buffers are allocated to each of the x8 ports, whereas six buffers are given to the x4 port and four buffers are available for the x1 port. Thus the amount of buffer available on a given port is dynamically assigned based on the traffic loading on each port, resulting in higher overall system performance.

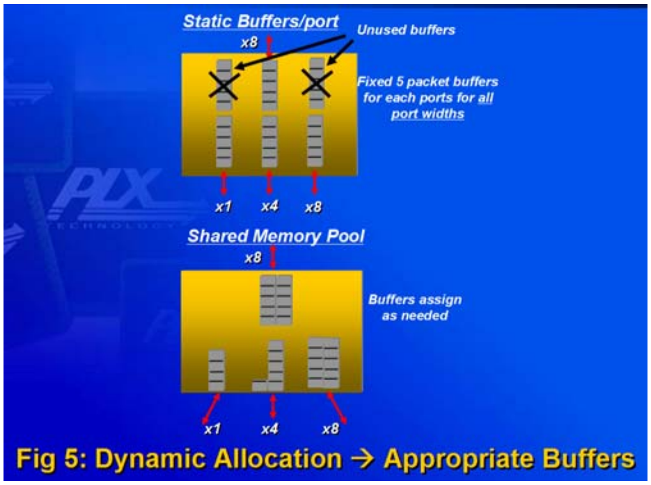
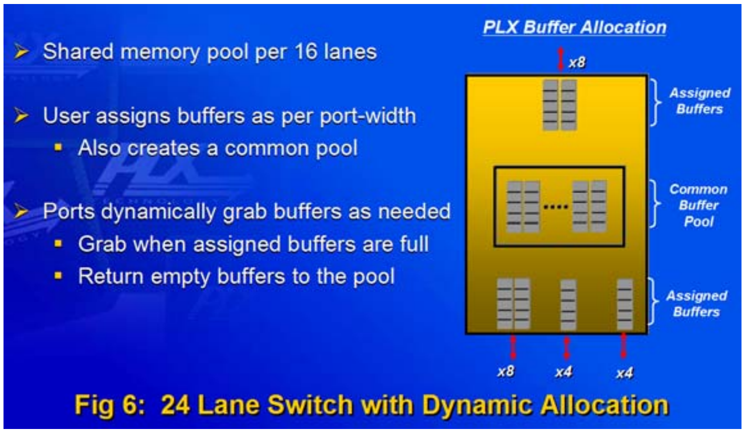


Fig 5: Dynamic Allocation → Appropriate Buffers

One major benefit of this architecture is faster UpdateFC packets. When the port runs out of credits in a fixed buffer system, it has to wait for the traffic to flow out of the buffer before it can release that buffer memory and send some more UpdateFC's to the sending device – this may require the device to wait for the destination device to also release credits. With dynamic buffering the traffic doesn't have to even leave the ingress queue before UpdateFC's are sent, because the switch can grab buffer capacity from the common pool and immediately send UpdateFC's. This results in superior performance especially, under bursty traffic conditions

Real-World Implementation of Dynamic Buffer Allocation

A real-world implementation of Dynamic Allocation can be seen in Figure 6. Here, a PLX PEX 8624 24-lane PCIe Gen 2 switch is configured with a x8 upstream port, a x8 downstream port and two x4 downstream ports. This switch's configuration has been set up by the user with assigned buffer space for each port and an uncommitted common (or shared) buffer pool per 16 lanes. The buffers have been assigned proportional to the port width, i.e., the x8 ports each have 10 packet buffers, the x4 ports four each. A common buffer memory pool is set up with five buffer packets for each of the 16 downstream lanes. Each of the ports may dynamically grab buffers as needed to support its own traffic bandwidth. Each of the ports may dynamically grab buffers without decoding which type of PCI Express traffic (Posted, Non-Posted, Completion") is selected in the incoming TLPs that have overflowed the first tier of credit memory, which has to decode these traffic types, as specified in the PCI Express specification. Having the second tier of credit memory un-decoded also improves performance, in cases where the initial credit advertisements don't match the traffic flows optimally. For example, a port may grab buffers when its assigned buffer memories are full; conversely, a port may return buffers to the pool when they are empty. This dynamic reallocation has two benefits in switch design: it makes full use of the buffer memory on-chip and it requires less overall memory to achieve optimal performance.



- Shared memory pool per 16 lanes
- User assigns buffers as per port-width
  - Also creates a common pool
- Ports dynamically grab buffers as needed
  - Grab when assigned buffers are full
  - Return empty buffers to the pool

Fig 6: 24 Lane Switch with Dynamic Allocation

Port Configuration Flexibility Improves Performance, Simplifies Layout

In the previous generations of PCIe switches, one port was fixed as the upstream port while all other ports were defined as downstream, with severely limited lane count/port count combinations. The PLX Gen 2 switch family offers flexible and versatile port configuration schemes, with ports configurable as x1, x2, x4, x8, and x16 for maximum port bandwidth ranging from 250MB/s (x1 port, Gen 1 signaling) to 8GB/s (x16 port, Gen 2 signaling), with several intervals in between. This means it is easier to optimize lane bandwidth and power dissipation and port layout trace-width from port to port. In addition, these new switches support auto-negotiation of the port width, reducing the number of lanes that are active in a port down to match endpoints that are connected. For example, if a NIC with a x4 port is connected to a x8 (or x16) port on the switch, the switch will automatically reduce the number of active lanes for that port down to a x4 configuration.

Selectable Upstream Port Simplifies High Performance Layout

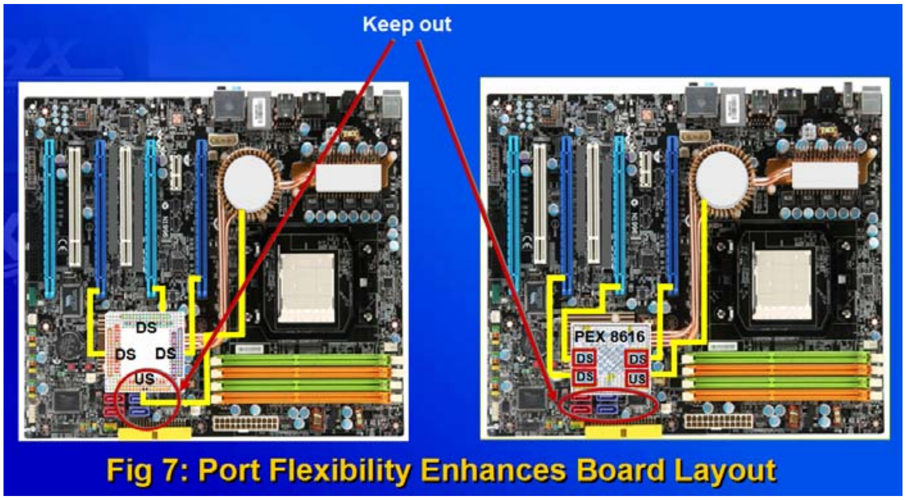


Fig 7: Port Flexibility Enhances Board Layout

The PLX Gen 2 switches also support a moveable upstream port. Any port, in fact, can be defined as the upstream port in these devices. This can be optimized to meet the needs of the traffic through each port of the switch. Additionally, the layout of a system board is enhanced by this flexible upstream port assignment. Figure 7 illustrates how, in a storage application, a flexible upstream port assignment allows spreading of high-speed traces evenly on a system board with a 16-lane switch configured with one four-lane upstream (US) port and three x4 downstream (DS) ports. The system board on the left uses a switch with a fixed US port. The fixed US port creates severe trace congestion since the DS ports are required to route through the SATA connectors, creating an undesirable crosstalk environment. The system board on the right shows the same system with a switch that has a flexible US port. This flexibility allows the layout designer to avoid routing the high-speed PCIe lanes through the equally high speed SATA2 data paths, thus reducing crosstalk, enhancing signal integrity and improving transmission margin.

Dual Cast

In addition to balancing bandwidth and improved buffer allocation, the PLX Gen 2 family of switches supports Dual Cast, a feature which allows for the copying of data packets from one ingress port to two egress ports, allowing for higher performance in dual-graphics, storage, security, and redundant applications. Without Dual Cast, the CPU must generate twice the number of packets, requiring twice the processing power. Figure 8 illustrates how data packets from a source port (upstream) are sent to the destination port and a Dual Cast port without CPU activity. Figure 9 illustrates a typical application for a Dual Cast switch. In this example, data is dual-casted through the PEX 8612 switch, feeding a redundant storage array of Fibre Channel drives. The effect is to double the I/O performance without loading the CPU. In addition, there is optional TLP address translation on the copied (dual-casted) TLP, to support applications which require identically copied memory buffers, but with an added address offset or modification on them.

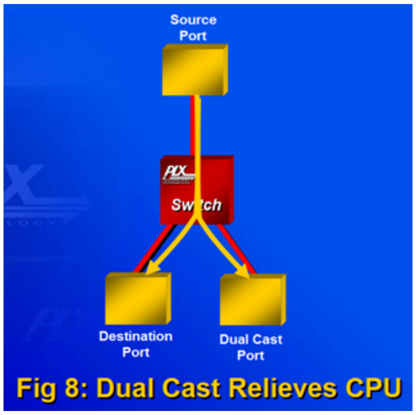


Fig 8: Dual Cast Relieves CPU

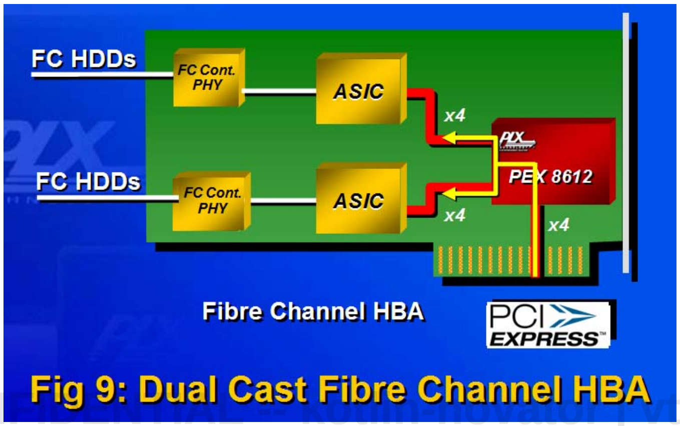


Fig 9: Dual Cast Fibre Channel HBA

Excellent Simulation Results

Figure 10 shows a Host-centric PCIe switch configured with five downstream ports using x8 links and one x8 upstream port. This simulation uses Memory Writes to the root complex.

Figure 11 plots the simulated performance of the topology in Figure 10. Throughput is charted as a function of payload size. As is expected, the best throughput is achieved with the larger payloads. In this chart, the white bars show the theoretical maximum throughput that is possible for one-way traffic on a x8 PCIe Gen 2 link. The blue bars show the PEX 8600 switch throughput. The red curve at the top of the chart shows the percentage of the maximum throughput that is achieved by the switch. This value ranges from 99.5% to 99.8%.

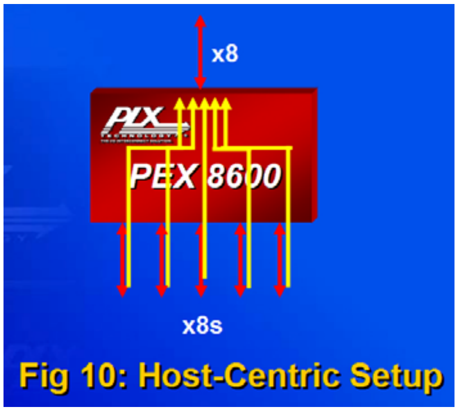


Fig 10: Host-Centric Setup

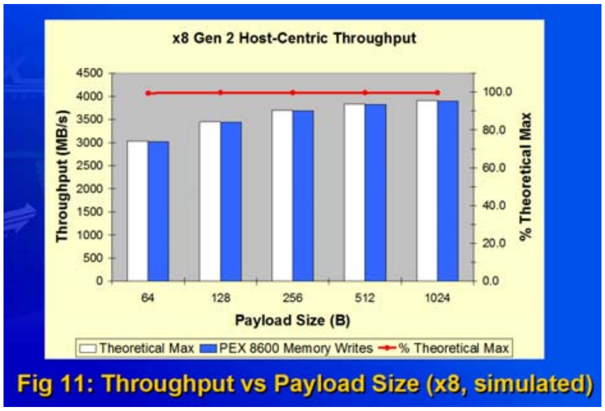


Fig 11: Throughput vs Payload Size (x8, simulated)

Figure 12 shows a peer-to-peer configuration using with one x16 Gen 2 upstream port and two downstream x16 Gen 2 peers. This simulation was performed with two-way symmetrical traffic.

Figure 13 charts the result. As is expected, the best throughput is achieved with the larger payloads. As in the previous chart, the white bars show the theoretical maximum throughput that is possible for one-way traffic on a x8 PCIe Gen 2 link. The blue bars show the PEX 8600 switch throughput with two-way traffic. The red curve at the top of the chart shows the percentage of the maximum throughput that is achieved by the switch. Again, this value ranges from 99.5% to 99.8%.

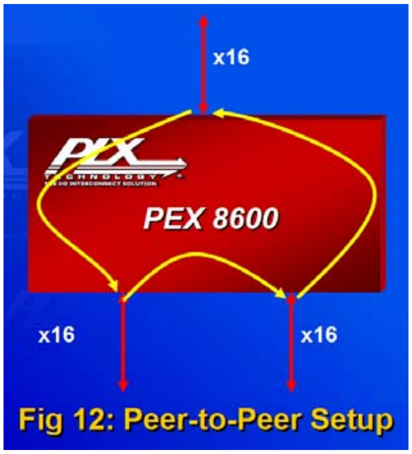


Fig 12: Peer-to-Peer Setup

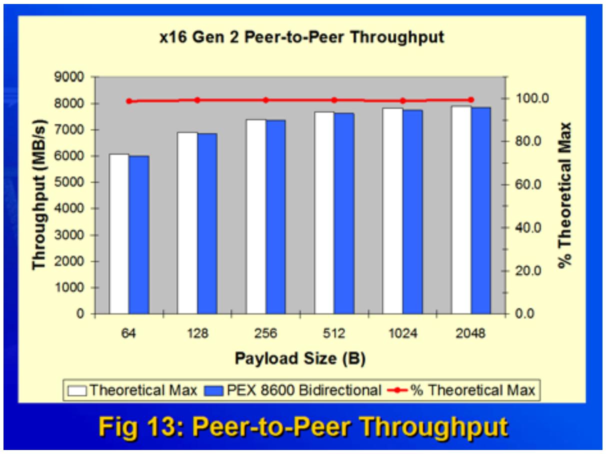


Fig 13: Peer-to-Peer Throughput

Figures 14 and 15 show a similar setup, using x8 links and seven ports instead of the three ports seen in the previous simulation example. In this simulation, the percentage of the maximum throughput that is achieved by the switch ranges from 99.5% to 99.3%.

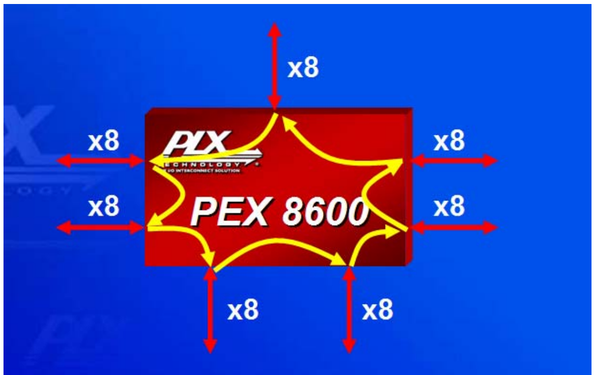


Fig 14: x8 Peer-to-Peer Setup

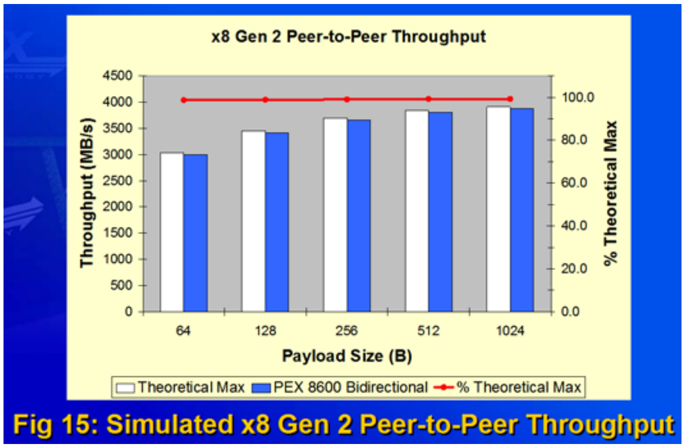


Fig 15: Simulated x8 Gen 2 Peer-to-Peer Throughput

Figures 16 and 17 show a similar setup, with x4 links. In this example, the red curve at the top of the chart, which indicates the percentage of the maximum throughput that is achieved by the switch, shows a slightly reduced value, which ranges from 97.9% to 98.7%.

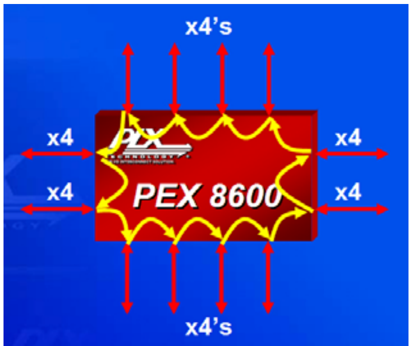


Fig 16: Simulated x4 Gen 2 Peer-to-Peer Throughput

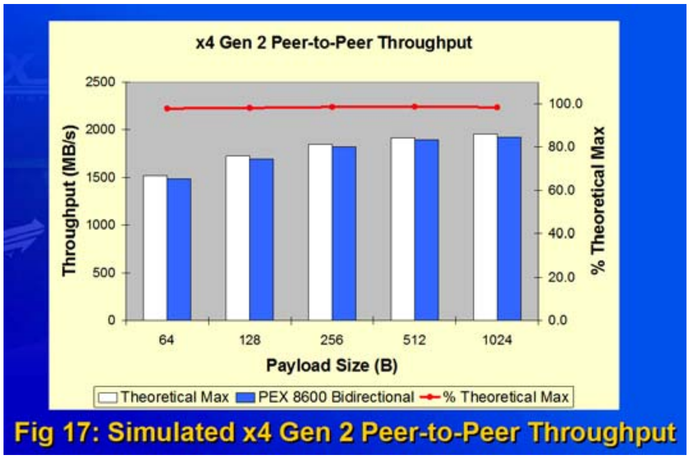


Fig 17: Simulated x4 Gen 2 Peer-to-Peer Throughput

Latency

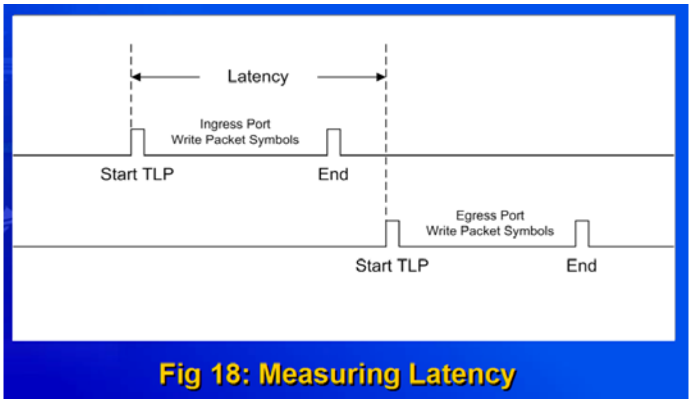


Fig 18: Measuring Latency

Figure 18 shows how latency is measured in these simulations. Figures 19-22 show simulated latency values for various ingress and egress port configurations, from x16 Gen 2 through x1 Gen 1. Latency is the length of time it takes to proceed from one event to another. Latency can be measured in several different ways, but perhaps the most common measurement for a switch is Start TLP-to-Start TLP (STP-to-STP) latency. Figure 18 illustrates an STP-to-STP latency measurement. When the Egress Start TLP symbol is transmitted out of a switch before the Ingress Port End symbol arrives, the transfer is termed Cut-Thru. If the Destination Port is not congested, the PEX 8648 always cuts the packet through. The PEX 8648 has the same latency, regardless of whether the traffic is upstream or peer-to-peer.

As expected with the PEX 8648 Cut-Thru architecture, STP-to-STP latency is basically constant for all Payload sizes, from any width to the same width or smaller, as indicated by the shaded-green entries in x. A faster Link can receive the Header for decode faster, with a slightly lower latency.

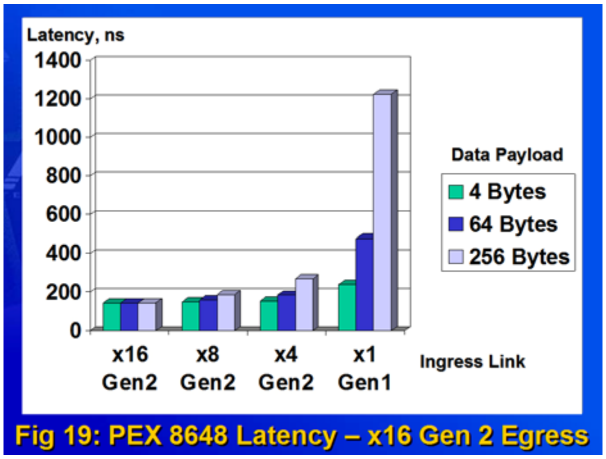


Fig 19: PEX 8648 Latency – x16 Gen 2 Egress

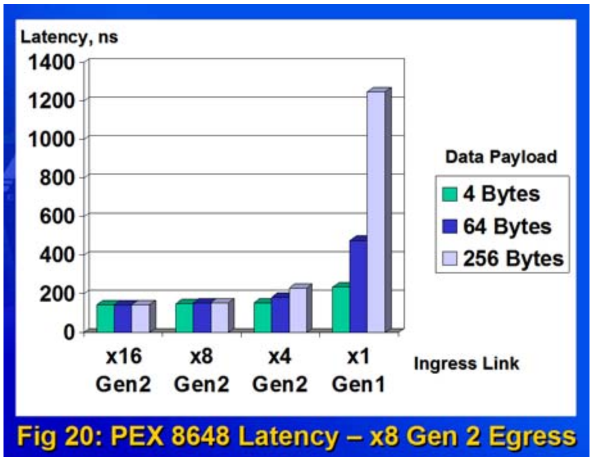


Fig 20: PEX 8648 Latency – x8 Gen 2 Egress

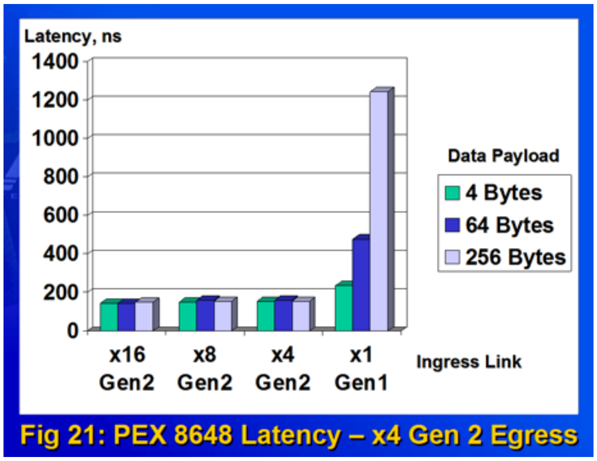


Fig 21: PEX 8648 Latency – x4 Gen 2 Egress

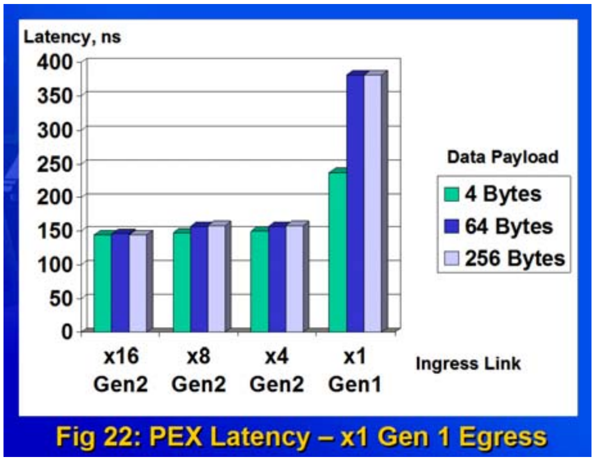


Fig 22: PEX Latency – x1 Gen 1 Egress

PLX Gen 2 Switches Approach Theoretical Maximum Throughput

Figure 23 compares simulated and measured throughput performance for the PEX 8624 switch, for three categories of data: 100% reads, 100% writes and 50/50% reads and writes. The measured values were averaged over large transfers (greater than several Mb). Third-Party Simulation Model delays, such as 1.8ns between write packets decreased throughput relative to theoretical max. Simulated values measured over read completions or write TLPs approximate large measured transfers. For the lab measurements, a Tylersburg Root Complex was used in a Gen 2 x8 configuration. The Endpoints were LeCroy Analyzers. For the Simulated Values, the Root Complex is the Avery RC Model and Avery Endpoint EP1 and EP2 Models were used to create simulated endpoints. As can be seen in the charts, the PEX 8646 achieves > 95% of the theoretical maximum throughput in all cases.

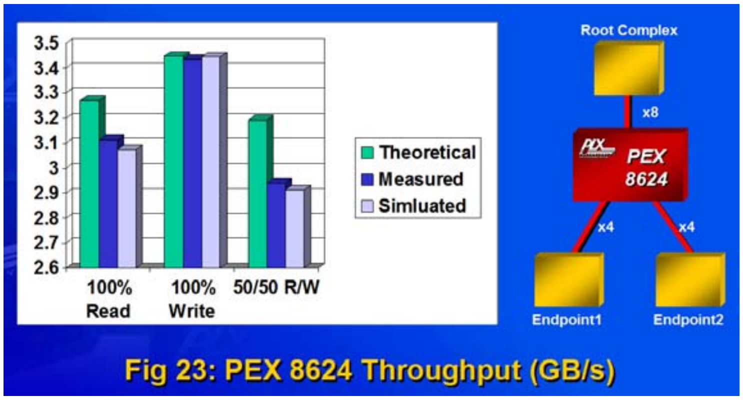
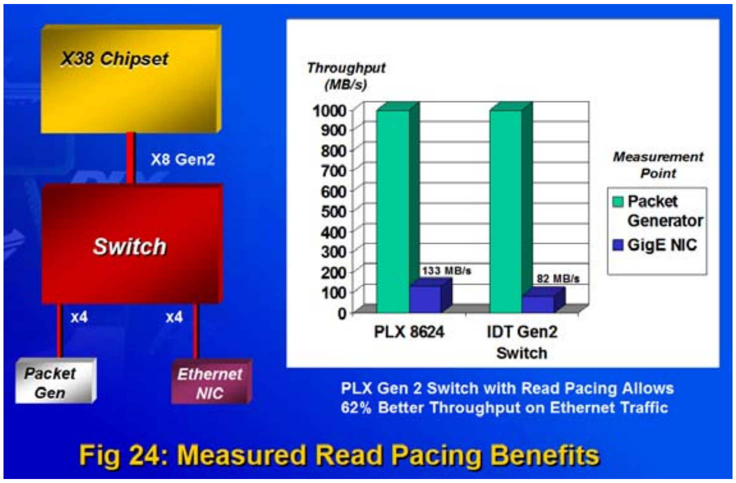


Fig 23: PEX 8624 Throughput (GB/s)

Real-World Read Pacing Measurements

Figure 24 illustrates the benefits of Read Pacing. One PLX Gen1 switch (PEX 8624) is configured with Read Pacing enabled. This allows the bursty traffic from the Gigabit Ethernet NIC to compete with the port- hogging packet generator, which is pumping a full Gigabyte/sec of x4 Gen 2 traffic. Another switch from IDT, which does not have Read Pacing technology available, is tried in the same configuration. As can be seen in Figure 24, the PLX switch with Read Pacing allows the GigE traffic 62% more throughput when compared to the IDT switch.



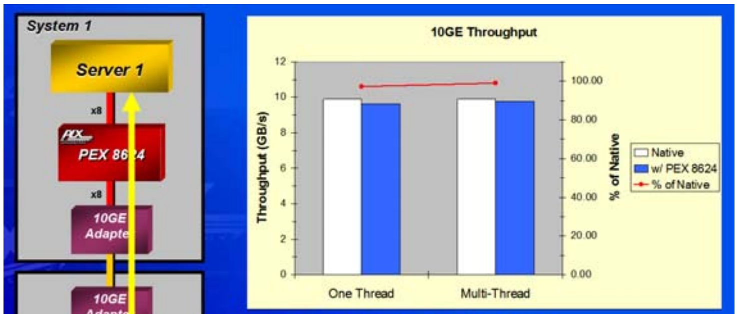
PLX Gen 2 Switch with Read Pacing Allows 62% Better Throughput on Ethernet Traffic

Fig 24: Measured Read Pacing Benefits

Throughput with 10GE Endpoint Exceeds 97% of Native

Figure 25 illustrates high bandwidth communication between two Quad Xeon servers via 10 Gig Ethernet. The PLX PEX 8624 switch provides >97% of native throughput in this example.

This example uses a Software-Based Ethernet Traffic Generator: Netperf, Iperf and a 10GE adapter are used to generate high speed streams through the switch. Gen 2 x8 links are used upstream and downstream on the switch.



- Ideal performance with PEX 8624
  - One Thread
    - 97.4% of Native Throughput achieved with PEX 8624
  - Multi-Thread (4 threads)
    - 99.0% of Native Throughput achieved with PEX 8624

Fig 25: PEX 8624 Measured Performance

Figure 26 shows a throughput test using the Spirent Test Center Traffic Generator system. Very close to 100% of Native Throughput achieved for all Frame Sizes. The >100% of Native Throughput is achieved for smaller frame sizes due to the buffer management between the switch and root complex.

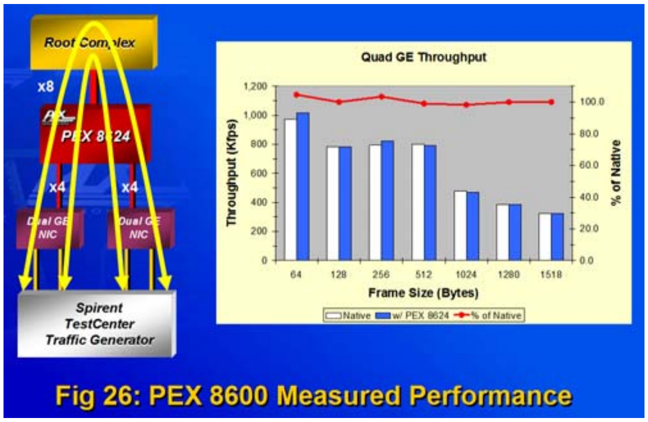


Fig 26: PEX 8600 Measured Performance

Summary

The PLX PEX 8600 series of Gen 2 PCIe switches provide extremely high-performance and low latency based on the PLX enhanced feature-set and unique switch architecture. This paper has provided lab measurements and simulation results that indicate that PLX switches provide almost zero constriction to high-speed traffic across all port configurations and payload sizes.