*Carter Buck, Sr. Applications Engineer, PLX Technology*

# 1. Introduction

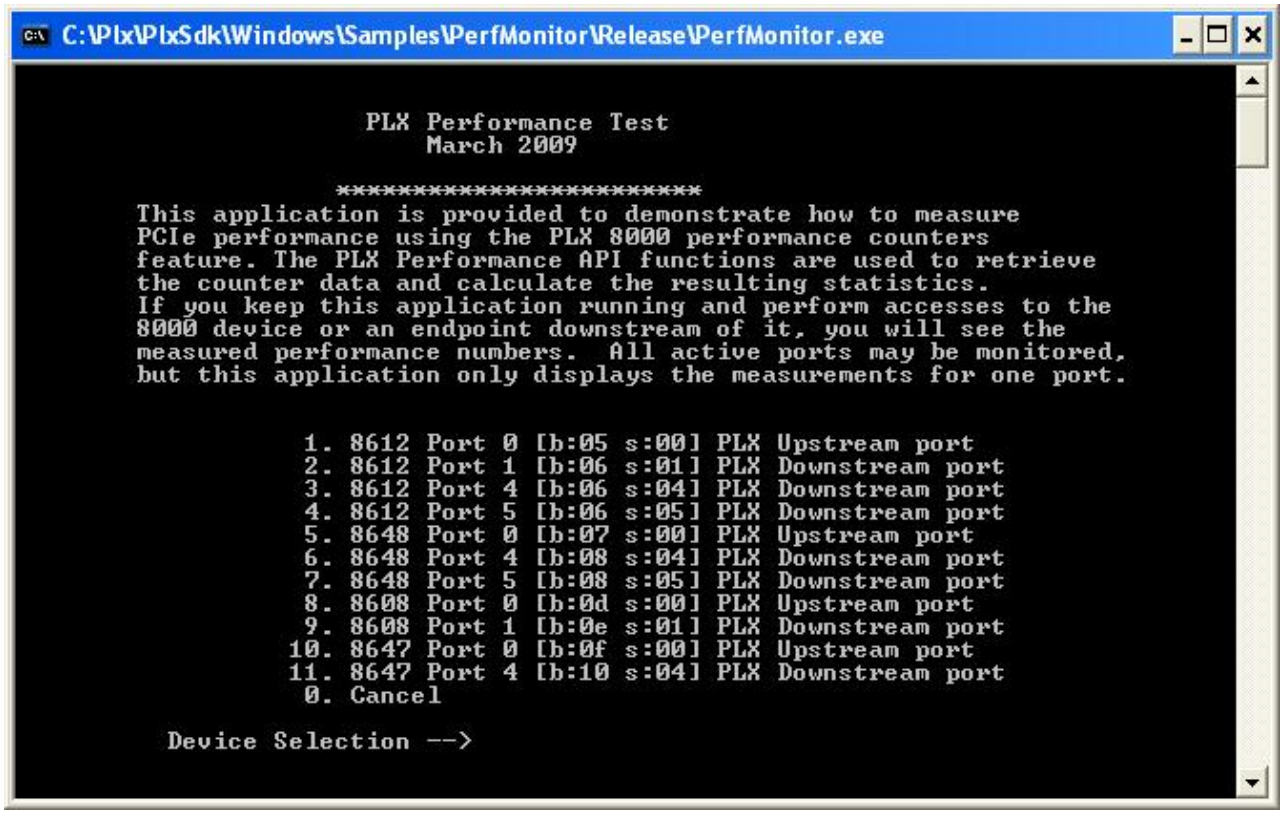
The PEX 8619 includes a Performance Monitor feature that provides benchmark numbers for performance analysis. To better understand throughput as it relates to the PLX switch, we need to know TLP and DLLP bandwidth on every port in both directions. For TLPs, we need to know the ratio of types of TLPs being used, and the average payload length of those TLPs. On-chip counters include:

- TLP header and payload counters (per TLP type, per Port, per direction)
- DLLP counters (per TLP type, per Port, per direction)

Software to support the Performance Monitor (including APIs with source and a sample application) is included in the PLX SDK. The PEX Device Editor also includes a GUI Performance Monitor application that additionally provides graphs showing performance over time.

# 2. Sample Application

The screen capture below shows an opening menu of the PerfMonitor sample application (listing installed PLX switches), and the path to the file within the installed PLX SDK.




This sample application in the PLX PEX SDK is located at:

C:\Plx\PlxSdk\Windows\Samples\PerfMonitor\Release\PerfMonitor.exe

Source for this sample application is located at:

C:\Plx\PlxSdk\Windows\Samples\PerfMonitor\PerfMonitor.c

# 3. Performance Monitor APIs

The PLX SDK includes five APIs to support the Performance Monitor feature:

PlxPci_PerformanceCalcStatistics
PlxPci_PerformanceGetCounters
PlxPci_PerformanceInitializeProperties
PlxPci_PerformanceMonitorControl
PlxPci_PerformanceResetCounters

These APIs are described in the PLX SDK User's Manual:
C:\Plx\PlxSdk\Documentation\PlxSdkUserManual.pdf

API source in the installed SDK is located at:
C:\Plx\PlxSdk\Windows\Driver\Source.PlxSvc\ApiFunc.c

# 4. Performance Monitor Registers

Three registers in Port 0 are used for the Performance Monitor:

568h – Monitor Control
618h – InOut Probe RAM Control
628h – Monitor Read FIFO

## 2.1 Monitor Control (568h)

| Bit | Value | Type | Name | Description |
|---|---|---|---|---|
| 25:0 | 0h | RW | Sample Count | |
| 26 | 00 | RsvdP | Reserved | |
| 27 | 0 | RW | Performance Monitor Start | Setting this bit starts the Performance Monitor |
| 28 | 0 | RW | Infinite Sample Enable | Setting this bit enables Continuous Sampling |
| 29 | 0 | RW | Factory Test Only | |
| 30 | 0 | RW | Monitor Reset | Setting this bit resets the Performance Monitor |
| 31 | 0 | RW | Monitor Enable | Setting this bit enables the Performance Monitor |

The Performance Monitor is started by Setting bits [31:30, 28:27] in a single Write (value 0xD8000000).
The Performance Monitor is stopped by Setting bit [30] (Write value 0x40000000).

## 2.2 InOut Probe RAM Control (618h)

| Bit | Value | Type | Name | Description |
|---|---|---|---|---|
| 0 | 1 | RWS | RAM Enable | Must be 1 to enable the RAM |
| 1 | 0 | RW1C | Reset RAM | Setting this bit resets the RAM |
| 2 | 0 | RW1C | Reset Read Pointer | Setting this bit resets the internal RAM pointer |
| 3 | 0 | RWS | Factory Test Only | |
| 5:4 | 10 | RWS | Capture Type | Must be value 10b for Performance Monitoring |
| 7:6 | 00 | RWS | Factory Test Only | |
| 8 | 0 | RWS | Factory Test Only | |
| 22:9 | 0h | RsvdP | Reserved | |
| 30:23 | 0h | RO | Factory Test Only | |
| 31 | 1 | RO | RAM Full Status | |

The Performance Monitor RAM is enabled by Setting bits [5, 1:0] (Write value 0x00000023). This RAM must be enabled each time the Performance Monitor feature is to be used.

### 2.3 Monitor Read FIFO (628h)

| Bit | Value | Type | Name | Description |
|---|---|---|---|---|
| 31:0 | 0h | RO | Monitor Read FIFO | Window to internal RAM |

The performance data is captured to internal RAM with a preset structure. Software repeatedly reads the Monitor Read FIFO register to retrieve the counter values. Each register read returns the next value from the sequence. The data structure is defined in the next section.

# 5. Performance Data Structure

The following page shows the data structure of the Monitor Read FIFO.

**Key**

```
*   IN    = Ingress port
*   EG    = Egress port
*   PH    = Number of Posted Headers (Write TLPs)
*   PDW   = Number of Posted DWords
*   NPDW  = Non-Posted DWords (Read TLP Dwords)
*   CPLH  = Number of Completion Headers (CPL TLPs)
*   CPLDW = Number of Completion DWords
*   DLLP  = Number of DLLPs
*   PHY   = PHY Layer (always 0)

    13 counters/port * 16 ports = 208 counters
```

| offset | Counter |
|---|---|
| 0 | Port 0 IN PH |
| 4 | Port 0 IN PDW |
| 8 | Port 0 IN NPDW |
| C | Port 0 IN CPLH |
| 10 | Port 0 IN CPLDW |
| 14 | Port 1 IN PH |
| : | : |
| 24 | Port 1 IN CPLDW |
| : | : |
| 12C | Port 15 IN PH |
| : | : |
| 13C | Port 15 IN CPLDW |
| 140 | Port 0 EG PH |
| 144 | Port 0 EG PDW |
| 148 | Port 0 EG NPDW |
| 14C | Port 0 EG CPLH |
| 150 | Port 0 EG CPLDW |
| 154 | Port 1 EG PH |
| : | : |
| 164 | Port 1 EG CPLDW |
| : | : |
| 26C | Port 15 EG PH |
| : | : |
| 27C | Port 15 EG CPLDW |

| offset | Counter |
|---|---|
| 280 | Port 0 IN DLLP |
| 284 | Port 2 IN DLLP |
| 288 | Port 4 IN DLLP |
| 28C | Port 6 IN DLLP |
| 290 | Port 8 IN DLLP |
| 294 | Port 10 IN DLLP |
| 298 | Port 12 IN DLLP |
| 29C | Port 14 IN DLLP |
| 2A0 | Port 0 EG DLLP |
| 2A4 | Port 2 EG DLLP |
| 2A8 | Port 4 EG DLLP |
| 2AC | Port 6 EG DLLP |
| 2B0 | Port 8 EG DLLP |
| 2B4 | Port 10 EG DLLP |
| 2B8 | Port 12 EG DLLP |
| 2BC | Port 14 EG DLLP |
| 2C0 | Port 1 IN DLLP |
| 2C4 | Port 3 IN DLLP |
| 2C8 | Port 5 IN DLLP |
| 2CC | Port 7 IN DLLP |
| 2D0 | Port 9 IN DLLP |
| 2D4 | Port 11 IN DLLP |
| 2D8 | Port 13 IN DLLP |
| 2DC | Port 15 IN DLLP |
| 2E0 | Port 1 EG DLLP |
| 2E4 | Port 3 EG DLLP |
| 2E8 | Port 5 EG DLLP |
| 2EC | Port 7 EG DLLP |
| 2F0 | Port 9 EG DLLP |
| 2F4 | Port 11 EG DLLP |
| 2F8 | Port 13 EG DLLP |
| 2FC | Port 15 EG DLLP |
| 300 | Port 0 PHY |
| : | : |
| 33C | Port 15 PHY |