



## Lattice PCI Express Throughput Demo

---

**User's Guide**

## Lattice PCI Express Throughput Demo Overview

### Introduction

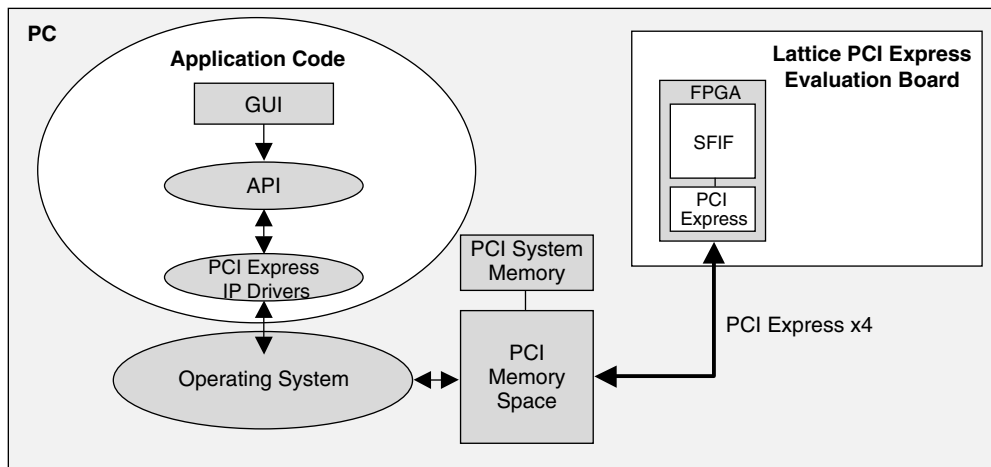
This user's guide describes how to run the Lattice PCI Express Throughput demo on a Windows system (Microsoft Windows2000, Windows XP SP2, Server2003). This demo uses the Lattice PCI Express IP Core to show the performance of the PCI Express system from the endpoint to the system memory. See the *Lattice PCI Express Demo Installation Guide* (UG05) for details on installing the demo software, evaluation board and Windows device driver.

### Demo Operations Overview

The purpose of this demo is to show the performance of the Lattice PCI Express SERDES hardware and PCI Express IP core when operating in a PC PCI Express expansion slot. The data rates for writes to the PC system memory, and reads from the PC system memory are measured and displayed in a graphical user interface. The demo performs Direct Memory Access (DMA) operation by transferring data directly to/from the PC memory. The demo uses an IP block named the SFIF (Stored FIFO InterFace) to generate read and write TLPs that will access the PC system memory. The SFIF exercises the PCI Express IP core and link with low overhead so the true performance of the PCI Express core and link can be measured.

The PCI Express interface is used for both control plane and data plane traffic. The control plane loads the SFIF memory and sets up the transfer. The data plane transfers the data from the SFIF to the PC memory. Figure 1 shows the relationship of the hardware and software components of the demo.

**Figure 1. PCI Express Throughput Block Diagram**



The demo software allows the user to set up different types of data transfers to understand the PCI Express link. The user can select the type of transfer to perform (write, Read, write/read, etc.) as well as how many bytes of data to transfer. The user also has the option of selecting the size of the TLP in which to perform the transfer.

### Background Knowledge

This demo assumes the user is familiar with basic PCI Express technology and is comfortable installing new hardware and software packages in a PC. Some experience in these areas is helpful to install the evaluation board and the software. Read the *Lattice PCI Express Demo Installation Guide* (UG05) for details on installing the evaluation boards, demo software and Windows device driver. A copy of the *PCI Express IP User's Guide* is also recommended.

Some recommended reference documents are:

- IPUG63, *LatticeECP2M x4 Endpoint IP User's Guide* (for LatticeECP2M users)
- IPUG57, *LatticeSCM PCI Express x1, x4 Core User's Guide* (for LatticeSCM users)

- EB22, *LatticeECP2M PCI Express x4 Evaluation Board User's Guide* (for LatticeECP2M users)
- EB31, *LatticeSC PCI Express x4 Evaluation Board User's Guide* (for LatticeSCM users)
- UG05, *Lattice PCI Express Demo Installation Guide*

## Installation Guide

Refer to the *Lattice PCI Express Demo Installation Guide* (UG05) for details on installing the evaluation boards, demo software and Windows device driver.

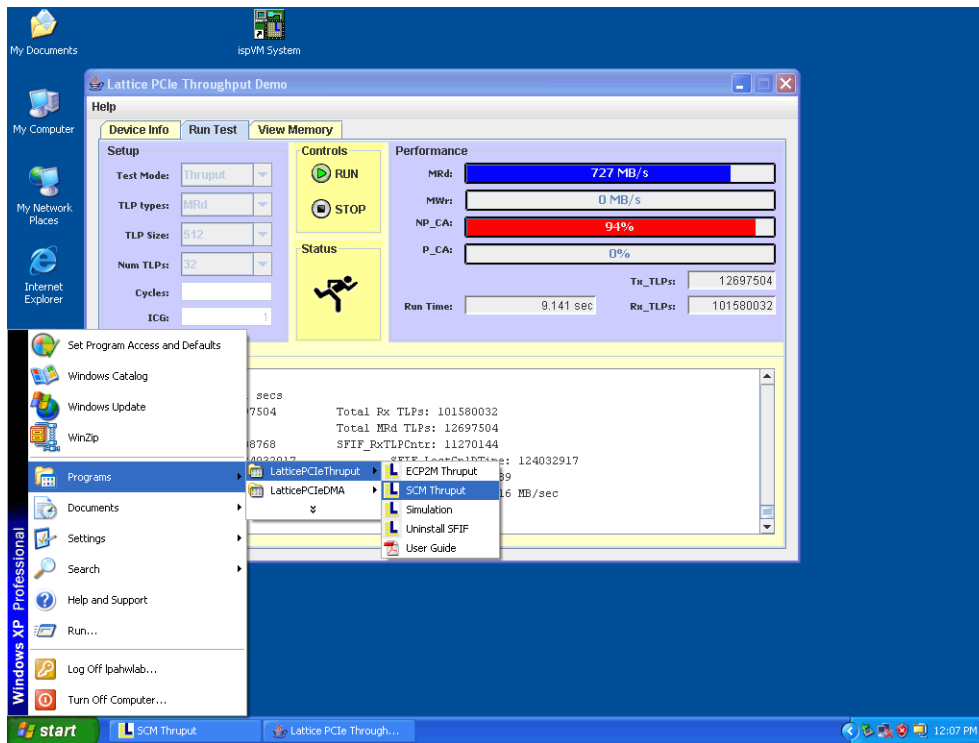
**NOTE: LATTICE IS NOT LIABLE FOR ANY LOSS OF DATA OR DAMAGES THAT MAY RESULT FROM THE INSTALLATION OF THE HARDWARE AND EXECUTION OF THE DEMO SOFTWARE. DO NOT INSTALL AND OPERATE ON MISSION-CRITICAL SYSTEMS.**

The Lattice PCI Express Throughput Demo package is released as a Windows executable installation file. The package includes the Demo GUI, device driver, bitstreams and documentation.

## Running the Demo

An operating PCI Express Throughput Demo is shown in Figure 2. The demo is started via normal Windows methods: desktop icons or through the Start Menu. The demo computes the throughput of the PCI Express link and displays the transfer rates with bar graphs. The user can select read, write and read-write throughput tests.

**Figure 2. Lattice PCI Express Throughput Demo**



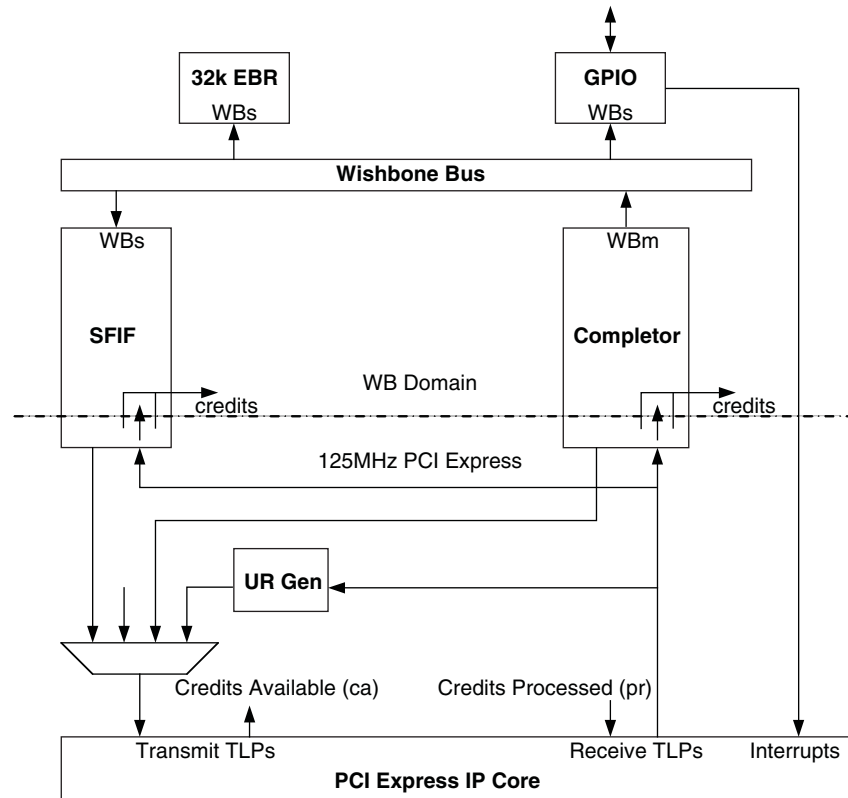
## Running the PCI Express Throughput (SFIF) Demo

Use the Start Menu (or desktop icons) to launch the LatticeSC or LatticeECP2M Throughput demo. There is no other setup required.

## SFIF IP Overview

Figure 3 shows a high-level view of the IP blocks within the Lattice FPGA

Figure 3. IP Blocks Block Diagram



The SFIF IP design allows the user to use the PCI Express link as both the control plane and data plane. For the control plane 1DW Memory Requests can be made through the Completor block. This block converts PCI Express TLP Memory Requests into Wishbone bus cycles. Both BAR0 and BAR1 are mapped to the same address space as defined below.

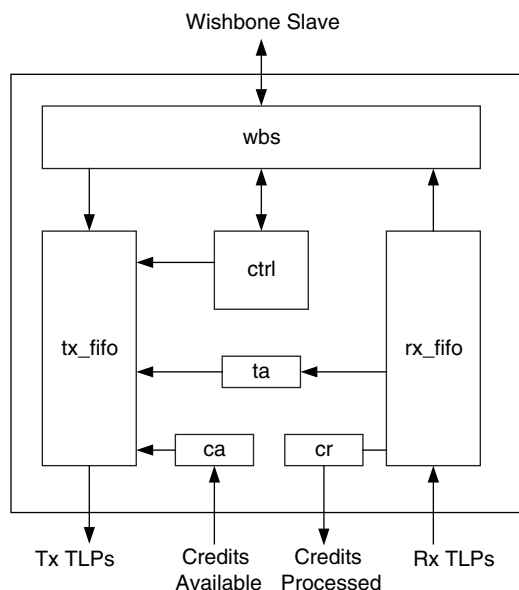
Table 1. BAR0/BAR1 Memory Map

BAR0/1 Address (Hex)	Wishbone Bus
	Device Description
0x00000000	GPIO
0x00010000	SFIF
0x00040000	EBR

The Completor block can be used to load the SFIF and exercise the data plane.

The SFIF details are shown in the following figure.

Figure 4. SFIF IP Details



The SFIF is made up of the following blocks.

- **Wbs** – This module is responsible for transactions on the wishbone bus.
- **Tx\_fifo** – This module stores up to 16KBs of TLPs loaded from the wbs to send over the PCI Express link
- **Rx\_fifo** – This module stores completions from the PC memory from read requests. It is 16kB deep.
- **Ctrl** – This module includes a state machine to perform all of the SFIF features such as looping through the tx\_fifo and providing an inter TLP gap (IPG).
- **Ca** – This module is responsible for checking if credits are available to send the current TLP.
- **Ta** – This module is responsible for checking if the tag is available for the next read request.
- **Cr** – This module is responsible for freeing CPL credits once completions are received.

The SFIF works by loading TLPs into the tx\_fifo via the wishbone bus. Once the TLPs are loaded into the tx\_fifo the user can determine how many times to iterate through the tx\_fifo (cycles) and how much gap to provide between the loops (IPG). The user then runs the SFIF and plays out the tx\_fifo over the PCI Express link. The SFIF transmitter will honor the credits available reported by the PCI Express core. It will not use an outstanding request's TAG value on a read request.

The rx\_fifo will monitor the receive bus of the PCI Express link waiting for completion TLPs with data. These TLPs will be stored in the rx\_fifo. As TLPs are stored in the FIFO, CPL credits are released to the system allowing the system to send more completions. Each CPLD TLP is stored along with a timestamp to mark the latency of read operations.

As shown in Table 2, the SFIF configuration registers are accessed at base address 0x1000 on the Wishbone Bus. The GUI software will address the registers in this range to setup the configuration registers, load the SFIF tx\_fifo and read the SFIF rx\_fifo.

**Table 2. SFIF Register Map**

Name	Addr (hex)	Width	Access	Description
Control	0	32	RW	Controls operation of SFIF
Tx TLP Control	4	32	W	Tx TLP control of loading TLP into FIFO
Tx IPG Control	8	16	RW	Tx Inter TLP Gap control
Tx TLP Data	C	32	W	DW of TLP data
Rx TLP Data	10	32	R	DW of Completion TLP data
Elapsed Count	14	32	R	Number of clock cycles in current run
Tx TLP Count	18	32	R	Number of Tx TLPs sent in current run
Rx TLP Count	1C	32	R	Number of CplDs received in current run
Credit Wait P	20	32	R	Number of clock cycles waiting for Posted Credits in current run
CplD Timestamp	24	32	R	Timestamp for last CplD
Credit Wait NP	28	32	R	Number of clocks cycles waiting for Non-Posted Credits or Tags in current run

The remaining IP in the demo design is used as simple testing blocks and are not part of the demo. The GPIO block is used to read the DIP switches and control the 16-segment LEDs. It also has the ability to generate an interrupt via the PCI Express core. The 32K EBR is again simply used for testing data movement via the completer block and is not part of the demo.

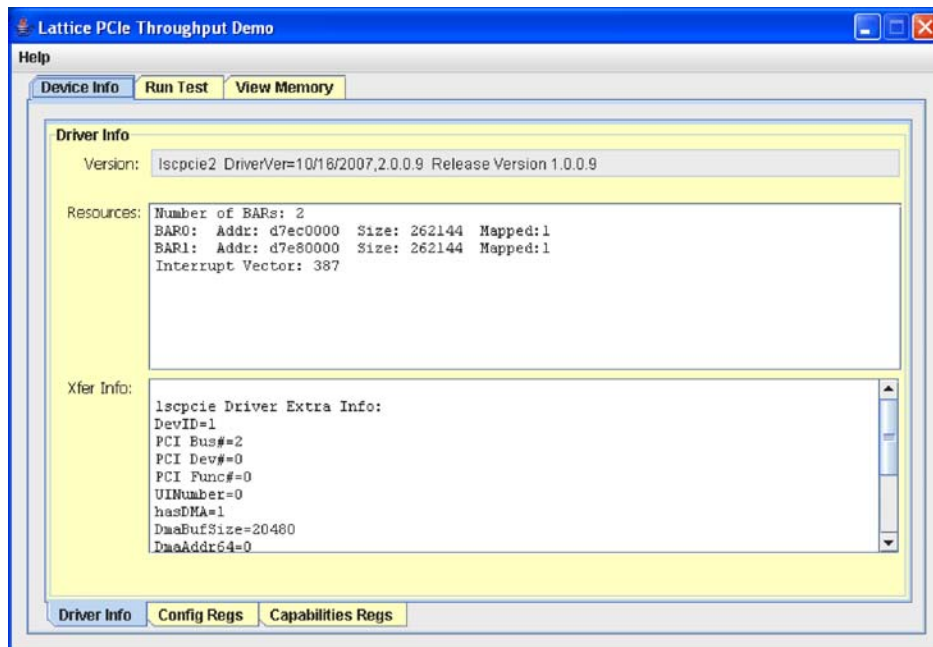
### Demo Software

This section explains the purpose and operation of the controls on the GUI screen.

### Device Info Page

The Device Info page is first displayed when the GUI starts. This information shows the evaluation boards PCI config space registers, PCI Express capabilities, and root complex buffer sizes. The Device Info page is displayed on three tabs, Driver Info, Config Regs, and Capabilities Regs.

**Figure 5. Device Info Page**



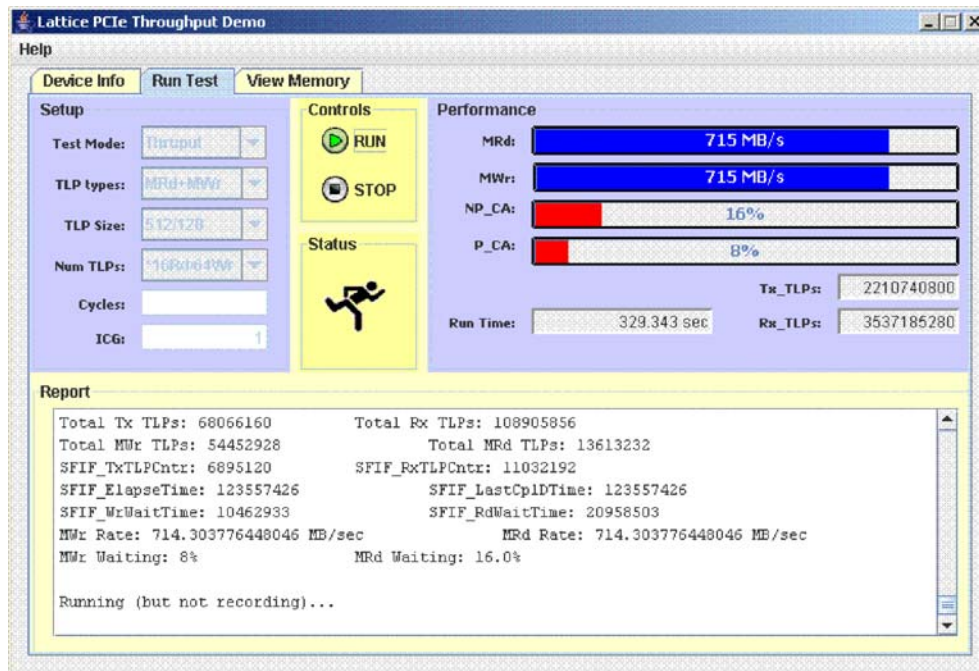
- **Driver Info Tab:** This page provides information about the device driver including the version, the resources used, and the transfer information. The demo design uses the Iscpici2 driver. The demo requests two BARs and a single interrupt vector. The Xfer Info box provides the buffer sizes for the root complex for Posted and Non-Posted TLPs. This information is important when considering the amount of credit waiting the demo design demonstrates when running a transfer. A root complex with larger buffers will provide better performance when running the demo since it will not have to release credits as quickly to allow the next TLP.
- **Config Regs Tab:** This page provides the standard PCI Type0 space configuration register contents. Things such as Device ID and Vendor ID are displayed and the assigned BARs.
- **Capabilities Regs Tab:** This page provides the link list of capability structures and their contents. Key information found in this box is the maximum TLP size supported by the root complex and the negotiated link width.

### Run Test Page

This page operates the demo design. This page has four parts:

1. Setup – Sets the configuration for the specific test
2. Controls – Start and stops the test
3. Performance – Displays the current data rates and other statistics
4. Report – Logs all of the test information

Figure 6. Run Test Page



The following controls exist on the Run Test Page.

- **Test Mode:** There are two modes of operation, cycles and throughput.
  1. Throughput - This mode allows the test to run continuously looping through the tx\_fifo and updating the performance numbers every second. This test will run until the user presses the STOP button.
  2. Cycles - This mode allows the user to set up a specific number of times the tx\_fifo will be looped. Once complete, the test will stop automatically and the performance numbers will be displayed based on the

---

entire run. The cycle consists of all MRd TLPS or all MWr TLPs. The purpose is to validate that the correct number of TLPs was sent/received using the counters and View Memory page.

The key difference between the two modes is how the performance data is displayed. A throughput test will provide new performance data every second. A cycle test will provide performance data after the number of cycles completes, or one second, whichever comes first. Note that cycles will not run longer than one second. Cycles tests run once; throughput tests run continuously until stopped.

- **TLP Types:** There are four types of TLP types which impact the type of “traffic” sent over the PCI Express link.
    1. MWr – Memory Write TLPs to write data from the endpoint to the PC system memory.
    2. MRd – Memory Read TLPs to read data from PC system memory to the endpoint.
    3. MRd+MWr – Both Memory Read and Memory Write TLPs are sent to the root complex.
    4. R+W+Ctl – Read, Write, and Control data are present on the PCI Express link. The Read and Write TLPs are sent from the SFIF while the PC is also modifying the GPIO 16-segment display LEDs. This TLP type shows both data and control plane TLPs sharing the PCI Express link.
  - **TLP Size:** The TLP size controls allows the user to select the size of the TLPs to be sent from the SFIF. The maximum size of the TLP will be dependent on the root complex. In MRd mode, the TLP size is limited by the Max. Read Request size (512 bytes). In MWr mode, the max TLP size is limited by Max. TLP Size (128 bytes). In Read/Write mode the following sizes are available for MRd TLP and MWr TLP combinations.
    1. 512,128 – 512-byte read requests with 128-byte write TLPs
    2. 256,128 – 256-byte read requests with 128-byte write TLPs
    3. 128,128 – 128-byte read requests with 128-byte write TLPs
    4. 64,64 – 64-byte read requests with 64-byte write TLP
    5. 32,32 – 32-byte read requests with 32-byte write TLPs
    6. 16,16 – 16-byte read requests with 16-byte write TLPs
  - **Num TLPs:** This control allows the user to select the ratio of read requests to write TLPs.
    1. 1Rd,1Wr – This ratio results in one Read Request and 1 Write TLP back-to-back. The completion data will need to be received before another Read Request can be made.
    2. 1Rd,4Wr – This ratio results in one Read Request and four Write TLPs. The completion data must be received before another Read Request can be made. This results in much greater bandwidth since read requests are four times the size of a write TLP. This ratio balances the PCI Express link, but still waits for read data.
    3. 4Rd,16Wr – This ratio results in four Read Requests and 16 Write TLPs. This ratio allows for four read requests to be outstanding (TAGs). This ratio is only recommended on server class motherboards due to the high bandwidth required. With four reads outstanding the root complex can better utilize the read data.
    4. 16Rd,64Wr – This ratio results in 16 Read Requests and 64 Write TLPs. This ratio allows for 16 read requests to be outstanding (TAGs). This ratio is only recommended on server class motherboards due to the high bandwidth required. With 16 read outstanding the root complex can better utilize the read data.
  - **Cycles:** This control is only available when the Cycle mode has been selected. This controls the number of times the tx\_fifo is looped before ending the test. The software starts the SFIF and waits one second while the SFIF transfers data (number of cycles). After one second, the software stops the SFIF and displays the performance. This has the effect of limiting cycles tests to a maximum of one second of operation. The cycles value cannot be larger than 65535 (it is a 16-bit counter).
-

In Throughput mode this control is not used. In Throughput mode the SFIF is looping the tx\_fifo continuously until the user presses the STOP button.

- **ICG (Inter Cycle Gap):** This control sets the number of 125MHz clock cycles between cycles. This control allows for modeling TLP traffic patterns that may be appropriate for a user's system. The ICG value cannot be larger than 65535 (it is a 16-bit counter).
- **Performance:** Data rates are displayed as progress bars, with the rate (MB/sec) displayed in the bar. The bars are updated every second when running in the Throughput mode or upon completion in the Cycle mode. The rates are computed from the hardware counters in the SFIF.

Write rates are computed from the following SFIF hardware counters: Tx TLP Count and Elapsed Count.

Write Rate (MB/sec) = (Tx TLP Count \* TLP Size) / (Elapsed Count \* 8ns)

Read rates are computed from the following SFIF hardware counters: Rx TLP Count and CplD Timestamp.

Read Rate (MB/sec) = (Rx TLP Count \* RCB\_Size) / (Elapsed Count \* 8ns)

RCB\_Size is the size in bytes of a CplD.

In Throughput mode the performance is recalculated every second and the counters are reset. In Cycles mode the performance is calculated once at the end of the run displaying the results for the entire transfer.

- **NP\_CA and P\_CA:** Time spent waiting for the root complex to accept TLPs is computed and displayed as a bar graph in percentage of time waiting over time spent running. Normal efficient operation should show a small percentage of time spent waiting for credits and more time spent sending TLPs.

Counters record when the SFIF wants to send a MRd but the credit available ports of the PCI Express core indicates the root complex has not yet processed the read requests. The PCI Express core is waiting to accept an UpdateFC-NP freeing up Non-Posted credits to send another MRd TLP.

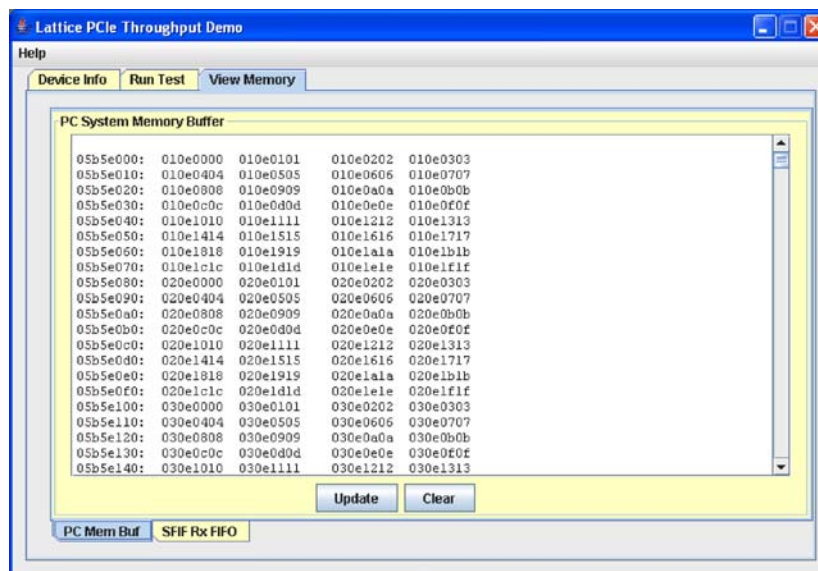
Counters record when the SFIF wants to send a MWr but the credit available ports of the PCI Express core indicates the root complex has not yet processed the sent write TLPs. The PCI Express core is waiting to accept an UpdateFC-P freeing up Posted credits to send another MWr TLP.

- **Report:** The Report box provides details about the test. In the Throughput mode this report will be updated every second up to 10 seconds. After 10 seconds the data is no longer updated in the report box to prevent system load and excessive resource usage during long duration tests (over night). In the Cycles mode the report window is updated when the test is complete.

### **View Memory Page**

The View Memory page allows the user to inspect the contents of the PC's system memory buffer allocated in the kernel space by the driver and is used for the source of MRd requests and destination for MWr TLPs. This can also be used to verify the MWr TLPs have worked and that the data was transferred from the PCI Express evaluation board into system memory.

Figure 7. View Memory Page



The SFIF Rx FIFO tab displays the parsed and formatted contents of the SFIF rx\_fifo. This can be used to verify that a small burst of MRd TLPs have returned the proper data to the evaluation board. The TLPs are parsed and time stamped.

## Frequently Asked Questions

### Q1. What does this demo show?

A1. This demo shows the Lattice PCI Express IP core's throughput capabilities. It demonstrates the core's throughput using a TLP generator to transmit TLPs to the root complex and store received TLPs in a FIFO. No run-time processing is done on the data. The purpose is to measure the maximum rates at which the core can send /receive TLPs. The data transfers are initiated by the evaluation board and written to and read from PC system memory, thus performing DMA transfers. This demo is not meant as a DMA reference design, but as a PCI Express throughput test.

### Q2. What is throughput? How is it different than bandwidth?

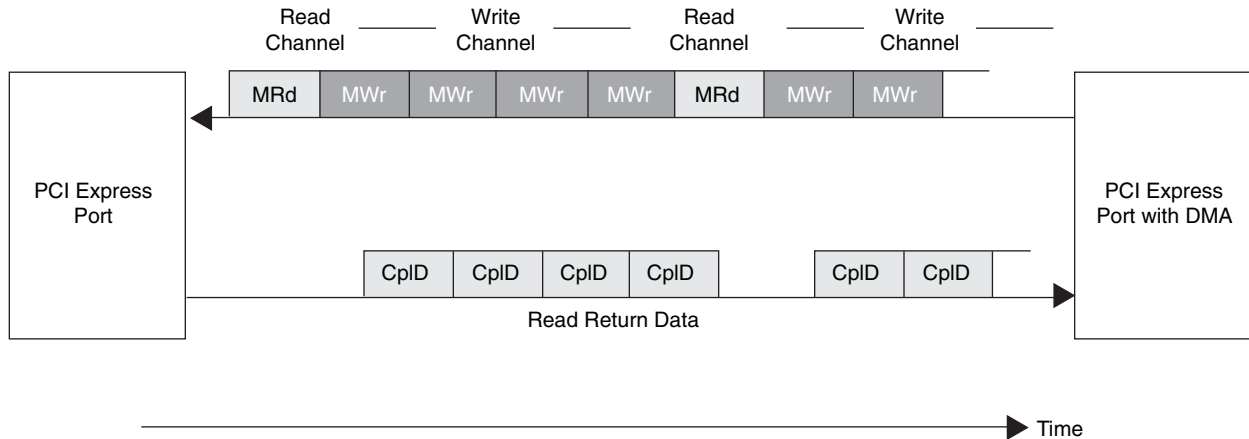
A2. Bandwidth is typically defined as the frequency at which a communication channel operates; the theoretical maximum rate of information transfer. Throughput is the amount of digital data per time unit that is delivered over a physical or logical link. A PCI Express x1 link operates at 2.5 GHz (bandwidth). A PCI Express x4 link operates at 4 \* 2.5GHz = 10GHz, which can be thought of as 10Gbits/sec. This is the bandwidth of the link. Significant overhead is present in PCI Express - 8b/10b encoding, skip order sets, update flow control packets, TLP headers, CRCs, etc. - all limit the actual throughput of user data to 85% of the bandwidth. Therefore, this demo assumes that 850MBps is the maximum throughput rate over a x4 link for the data payload, and that is what the bar graphs are scaled to. The rates are computed in MBps (1x10E6 bytes per second).

### Q3. What are TLPs?

A3. PCI Express performs packet-based data transfers (think IP packets) using Transaction Layer Packets (TLP). Three types of "User Data" packets are transferred during this demo:

1. Memory Write TLPs (MWr)
2. Memory Read TLPs (MRd)
3. Completion With Data (CpID)

Figure 8. Example TLP Types



Memory Write TLPs (MWr) are used to transfer data from the SFIF to the PC system memory. A MWr (in this demo) has the following format:

Figure 9. MWr Format

3DW Header (12 bytes)	Data Payload (4-128 bytes)	CRC (4 bytes)
--------------------------	-------------------------------	------------------

Memory Read Request TLPs (MRd) are used to request data from the PC system memory to be returned to the SFIF. A MRd can request up to 512 bytes to be read from the PC. A MRd (in this demo) has the following format:

Figure 10. MRd Format

3DW Header (12 bytes)	CRC (4 bytes)
--------------------------	------------------

Memory Read Completions with Data TLPs (CplD) are used to transfer data from the PC system memory to the SFIF in response to a MRd. A CplD can hold a maximum of 64 bytes of payload. Therefore multiple CplDs are sent in response to a single MRd request. A CplD (in this demo) has the following format:

Figure 11. CplD Format

3DW Header (12 bytes)	Data Payload (4-64 bytes)	CRC (4 bytes)
--------------------------	------------------------------	------------------

Each CplD has a tag value in the header. The tag corresponds to the tag given in the MRd request. The tag is used to associate returned data with the request for it. All outstanding requests must have unique tags assigned to them. In the demo, the Num\_TLPs field determines the number of tags that can be used, thus the number of simultaneous outstanding MRds. If the user selects MRd Throughput test with Num\_TLPs = 4, then tag numbers 1, 2, 3 and 4 will be sent and the SFIF will have to wait for all data (CplDs) for tag 1 to return before it can issue another MRd using tag 1 or any other outstanding tag (i.e., start another cycle of transfers).

**Q4. How is throughput computed?**

A4. Throughput is computed based on the payload portion of a TLP. The total number of transferred bytes is calculated from TLP size (payload in bytes) \* number of TLPs sent. There are two ways that the computation is performed:

1. Cycle Test Mode: Cycle mode uses the selected cycles and TLP sizes to compute the total payload transferred, and uses hardware counters to compute the time taken to perform the transfer.

$$a. \text{Wr\_Thruput} = (\text{MWr\_TLP\_Size} * \text{Num\_MWr\_TLPs} * \text{cycles}) / (\text{Elapsed\_clks} * 8\text{E-}9) / 1\text{E}6 \text{ MBps}$$

$$b. \text{Rd\_Thruput} = (\text{MRd\_TLP\_Size} * \text{Num\_MRd\_TLPs} * \text{cycles}) / (\text{LastCplDClk} * 8\text{E-}9) / 1\text{E}6 \text{ MBps}$$

2. Throughput Test Mode: Throughput mode estimates how many packets of each type have been sent or received and bases the throughput computation on these approximated numbers. The throughput test programs the SFIF control registers and TX FIFO, and runs the SFIF for approximately one second. At the end of one second, the SFIF is stopped and the counters are examined and the throughput computed as:

$$a. \text{Wr\_Thruput} = (\text{MWr\_TLP\_Size} * \text{Num\_Tx\_TLPs} * (\text{Num\_Wr\_TLPs} / (\text{Num\_Wr\_TLPs} + \text{Num\_Rd\_TLPs}))) / (\text{Elapsed\_clks} * 8\text{E-}9) / 1\text{E}6 \text{ MBps}$$

$$b. \text{Rd\_Thruput} = (\text{LastCplDClk} * 8\text{E-}9) / (\text{CplD\_Size} * \text{Num\_Rx\_TLPs}) / 1\text{E}6 \text{ MBps}$$

This method is not exact. The number of TLPs received is assumed to accurately track the number of CplDs returned per MRd requests. Also the CplD size is assumed to be 64 (valid for all current PC systems). The number of MWr TLPs is computed from the total number of TLPs transmitted from the TX FIFO. The total number also includes MRD request TLPs (if the MRd+MWr mode is selected). The number of MWr TLPs is based on the ratio of MRds to MWrs as selected in the Setup panel.

#### Q5. Why are MRd rates typically lower than MWr rates?

A5. PCI Express performs memory reads using split transactions. This means a memory read request is first sent, and at a later time, after the completer processes the request, the read data is returned. Memory writes, on the other hand, contain the data being written in the command. Memory reads also have to be split over multiple completions if the requested size is greater than the Read Completion Boundary (RCB) of the system, typically 64 bytes. Memory writes are therefore more efficient from an overhead standpoint because they require fewer packets to transfer the same amount of data.

#### Q6. Why are MRd+MWr rates typically lower than their individual tests?

A6. Two factors contribute to lowering the overall throughput.

1. Some of the MWr bandwidth now needs to be shared with sending MRd requests. The Tx pipe is now filled with both MWr and MRd TLPs, and not just MWr alone.
2. The SFIF can only transfer TLPs in the sequential order they are read from the Tx FIFO. As such, if a MRd blocks waiting for credits or tags, anything behind it (MWr) is also blocked (or a similar situation could occur if Posted Credit Available (P\_CA) is exhausted for MWr, then MRds would be blocked too. This is a deficiency of the SFIF design, and not a reflection of the PCI Express core. A better design would include separating the MRd and MWr FIFOs feeding the PCI Express core.

#### Q7. What does R+W+Ctl test?

A7. The Read+Write+Control test demonstrates that the data plane and control plane can both operate simultaneously through the PCI Express core. The SFIF performs the standard read/write data transfers to/from PC memory while at the same time the application software on the PC reads and writes registers in the GPIO block. The LEDs are blinked and the contents of fixed registers are checked. If an error occurs a warning sign is displayed on the GUI screen.

#### Q8. Why do performance numbers vary between machines/runs/slots?

A8. Different slots and different machines have different initial credits available. If a link has few credits available, then the performance will be lower because the SFIF will have to wait for the Root to process requests and free credits before more requests can be sent.

Performance can also vary between runs due to system loads. This appears to be the case in systems that have integrated video that shares PC system memory. Also, the application load on memory resources can affect

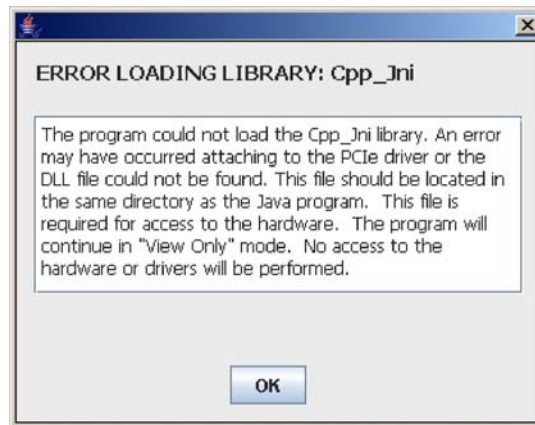
throughput because the CPU and the evaluation board are both accessing memory through the same port (Root Complex).

#### Q9. Can the demo crash my system?

A9. The demo performs direct memory access into PC system memory at high data rates (the maximum the PC can support). Under extreme load conditions, certain motherboards “hang” during MRd+MWr traffic with a large number of outstanding MRd requests (Tags).

#### Q10. The following error message is displayed when I start the demo. What does this mean?

**Figure 12. Error Loading Library Message**



A10. This error message indicates that the Java GUI could not link with the required DLL that implements the calls to the device driver (by default the installation process installs all required files in their correct locations) or, more likely, the GUI called the DLL to open the driver, but a corresponding board was not found in the system. There are two things to check:

1. Make sure that a board is installed and seen by the OS (see Troubleshooting section for more information)
2. Make sure you are executing the correct startup script for the correct board (i.e. the LatticeECP2M50 evaluation board is installed to run the LatticeECP2M demo).

#### Q11. What is the Simulation option?

A11. The Simulation executable (**Start->Programs->LatticePCIeThruput->Simulation**) starts the GUI in a mode that simulates transfers on hardware. The purpose is to allow the user to become familiar with the GUI pages and settings without requiring an evaluation board installed in the system. The simulation mode shows varying transfer rates for different test modes and traffic patterns, but these numbers are not representative of any real system. This is not a cycle-accurate hardware emulator. It is meant to be visually appealing, but do not base anything off observed rates or behavior in simulation mode.

## Troubleshooting

Refer to the Troubleshooting section in the *Lattice PCI Express Demo Installation Guide* (UG05) for details on diagnosing board, driver and software installation and operation issues.

## References

The following documents provide more information on topics discussed throughout this guide:

- IPUG63, *LatticeECP2M x4 Endpoint IP User's Guide* (for LatticeECP2M users)
- IPUG57, *LatticeSCM PCI Express x1, x4 Core User's Guide* (for LatticeSCM users)

- EB22, *LatticeECP2M PCI Express x4 Evaluation Board User's Guide* (for LatticeECP2M users)
- EB31, *LatticeSC PCI Express x4 Evaluation Board User's Guide* (for LatticeSCM users)
- UG05, *Lattice PCI Express Demo Installation User's Guide*

## Demo and System Requirements

- PC Motherboard with a x4 (or x8 or x16 compatible) slot
- Windows 2000, XP or Server2003 Operating System
- Lattice ispVM System, version 17 and JTAG download cable
- LatticeSCM80 PCI Express x4 Evaluation Board, SPI Flash loaded with PCIex4\_SC80\_SFIF\_15.bit
- LatticeECP2M50 PCI Express x4 Evaluation Board, SPI Flash loaded with PCIex4\_ECP2M50\_SFIF.bit
- Demo Install Package: LSC\_Win2kXP\_PCleThruput\_1\_0.exe
- Java Library Package (not required if previously installed): LSC\_Win2kXP\_JRE\_1\_5\_0.exe

## Technical Support Assistance

Hotline: 1-800-LATTICE (North America)  
           +1-503-268-8001 (Outside North America)  
 e-mail: techsupport@latticesemi.com  
 Internet: [www.latticesemi.com](http://www.latticesemi.com)

## Revision History

Date	Version	Change Summary
October 2007	01.0	Initial release.
January 2008	01.1	Updated Introduction, Demo Operations Overview, Background Knowledge and Installation Guide text sections.
		Updated Running the Demo section and added new screen shot.
		Updated Troubleshooting text section.

## Appendix A. Sample System Performance

Table 3 shows performance measured on three different PC systems. The Intel D975XBX and HP Proliant are server class machines, and the performance reflects this (note the large amount of credits). The ASUS is a consumer grade, desktop motherboard.

**Table 3. Sample System Performance**

System	Slots	RC Credits		Throughput (MBps)		
		PD(w)	NPH(r)	Read	Write	Read/Write
<b>Intel D975XBX</b>	x4	64	16	685	827	635/635
Intel 975/ICH7	x8	32	14	727	439	410/410
Windows 2000	x8	32	14	727	439	410/410
<b>ASUS P5RD1-VM</b>						
ATI						
Windows XP	x16	32	16	715	345	355/355
<b>HP Proliant DL145 Server</b>						
AMD						
Windows Server 2003	x8	64	32	727	827	715/715