



LatticeECP3 AMC Serial RapidIO Demo

User's Guide

Introduction

This document provides an overview of the LatticeECP3 AMC Serial RapidIO Demo running on the LatticeECP3 AMC Evaluation Board. The demo operates in a loopback mode, so no additional hardware is required to demonstrate the sending and receiving of RapidIO packets. A description of the HDL demo architecture is provided so that data flow during demo operations can be visualized.

Overview

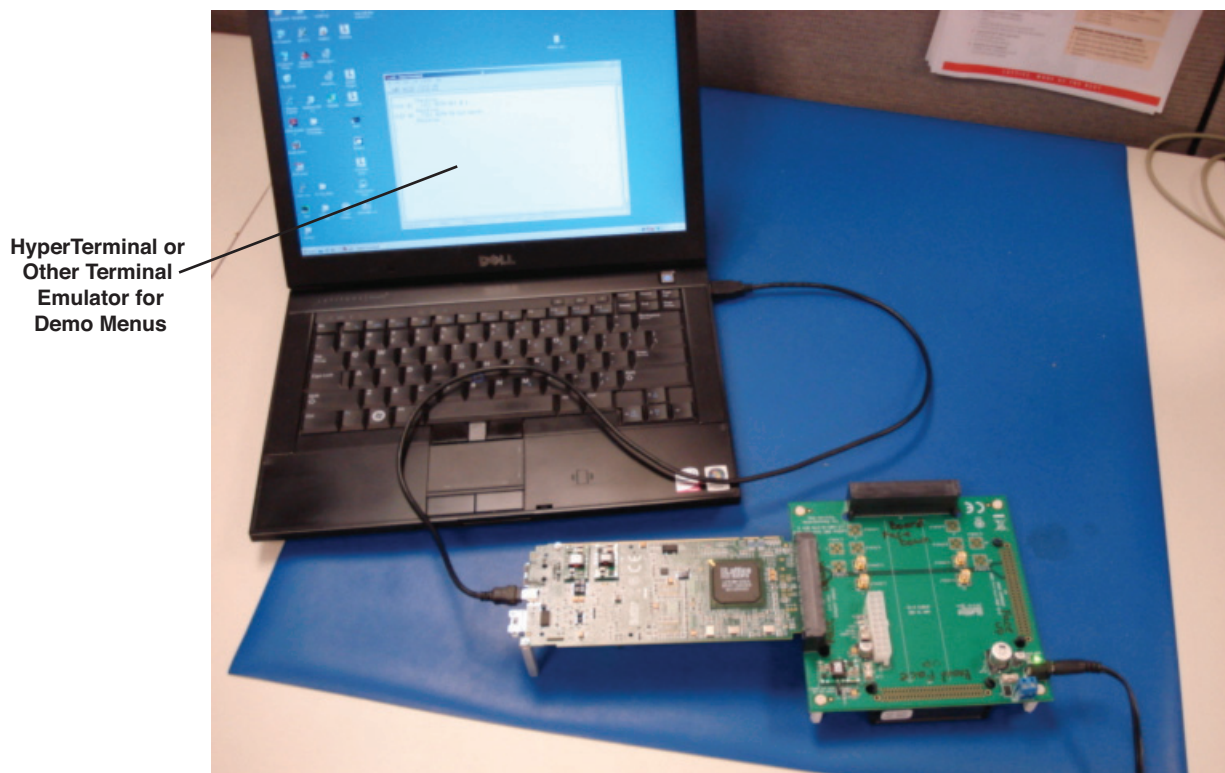
The LatticeECP3 AMC Serial RapidIO Demo provides a stand-alone platform to evaluate and test the Lattice RapidIO 2.1 Serial Endpoint IP core in a true hardware environment. The demo runs on the LatticeECP3 AMC Evaluation Board, an industry-standard AMC form-factor board. The LatticeECP3 AMC Evaluation Board is installed into the loopback slot of the Lattice AMC Interface Card to provide a stand-alone demonstration of the sending and receiving of RapidIO packets, along with inspecting RapidIO link configuration registers and displaying link statistics.

The user interacts with the demo through a text-based menu system provided by the on-chip LatticeMico32™ soft microprocessor. The LatticeMico32 is fully contained within the demo design in the FPGA. The software running on the LatticeMico32 provides the user interface, Serial RapidIO IP configuration and control, and RapidIO packet generation and decoding. The software performs various operations, such as initiating transactions, displaying link status, inspecting Serial RapidIO configuration registers and viewing test memory to confirm Serial RapidIO transactions change the addressed memory.

The only extra equipment needed is a PC, with a USB port, running terminal emulation software to view the menus and control the demo.

Figure 1 shows the LatticeECP3 AMC Serial RapidIO Demo set-up.

Figure 1. LatticeECP3 AMC Serial RapidIO Demo Set-up



The serial port provides the user interface. The serial port on the LatticeECP3 AMC Evaluation Board is implemented using a USB connection. A mini-USB cable, supplied in the demo kit, connects the board to a PC. The PC

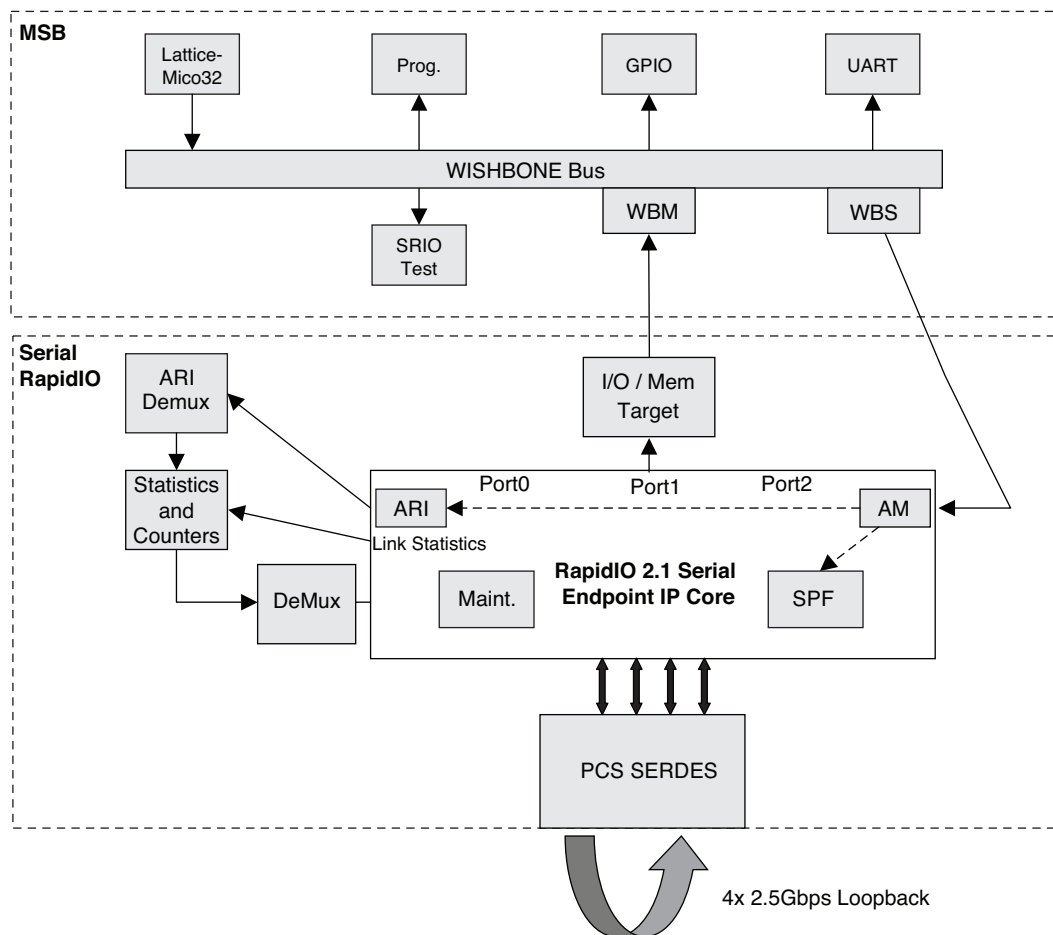
runs a USB to virtual Comm port program that allows the HyperTerminal to connect to the board over the USB cable.

Implementation

The purpose of the demo is to provide a working example of the RapidIO 2.1 Serial Endpoint IP core in hardware. As delivered, the demo operates in loopback mode, sending and receiving Serial RapidIO packets to itself. The demo settings can be reconfigured to allow communication with another Serial RapidIO endpoint. Interoperability is not the focus of this user's guide, but it is important to note that this is possible. This simple, self-contained demo operates in stand-alone loopback mode.

The architecture of the demo is presented as a reference point for the data flow during demo operations. Figure 2 shows the hardware architecture of the demo.

Figure 2. Demo Hardware Architecture



The primary components of the demo include the LatticeMico32 MSB processor platform and the Serial RapidIO sub-system. The Serial RapidIO sub-system contains the RapidIO 2.1 Serial Endpoint IP core and associated modules that interface the IP core to the PCS/SERDES and provide additional statistics counters.

The LatticeMico32 MSB processor platform runs the entire user interface software from on-chip program memory. The processor platform also contains the UART component that provides the serial port communications to the external PC. A WISHBONE Slave Port (WBS) provides access from the CPU bus to the Serial RapidIO IP registers. These registers are accessed through the Alternate Management Interface (AMI) port on the IP core. The AMI registers allow access to the standard Serial RapidIO configuration registers, as well as implementation-specific registers and features of the IP core.

The Soft Packet FIFO (SPF) block inside the IP core provides a mechanism to send and receive RapidIO packets under software control. To send a RapidIO packet, the software constructs the complete packet header and any associated payload, and loads the entire packet into the SPF. Once the SPF has been loaded with the packet to send, the SPF sends the RapidIO packet into the transport layer of the Serial RapidIO stack, as if the packet had been sent in from one of the logical layer ports. The packet is then serialized and sent out the SERDES channels. Since the SERDES channels are looped back onto themselves by the traces on the AMC Interface Card, the packet is received by the RapidIO 2.1 Serial Endpoint IP core.

Maintenance packets received at the core are directed to the internal Maintenance handling block. The Maintenance block handles all register reads and writes internally. Responses to Maintenance read requests are formulated and sent back entirely within the core with no user logic or software intervention.

Memory request packets received by the core are directed to Logical Port #1 by the DeMux policy. Attached to Logical Port #1 is a Target Memory module that converts a received RapidIO packet into a WISHBONE bus cycle. The WISHBONE bus cycle drives the LatticeMico32 CPU bus through the WISHBONE Master (WBM) port. A block of 64 kB of Serial RapidIO test memory is provided for the source/destination of Serial RapidIO memory transactions. This memory is also accessible by the CPU so software can verify Serial RapidIO transfers to the memory, or fill with a pattern to be read back via Serial RapidIO transfers.

Serial RapidIO memory requests are satisfied by the Target Memory module. Responses to an NRead or an NWrite_R are assembled by the Target Memory module and sent into the transport layer of the RapidIO 2.1 Serial Endpoint IP core. When the response arrives back at the core, the Demux policy routes the packet to the SPF since that was the initiator of the request.

Responses to a RapidIO packet (e.g., data requested from a Maintenance Read or NRead) are sent out and received back in the IP core. Received packets are directed to the SPF module. Routing is based on the upper bits of the SrcTID field. These upper two bits are set to 2'b11 to indicate that a packet was sent from the SPF and that responses should be directed back to the SPF. The DeMux policy block handles this decoding and routing of received packets.

The demo software monitors the SPF when expecting a response to be returned. Once the response arrives in the SPF, the software reads it and verifies the response type, status, tag, and contents. If the packets fails to return (times out) or is invalid, the software displays an error. Otherwise, the payload of the response and status are displayed to the user indicating a successful Serial RapidIO packet transfer.

The demo software also uses the Serial RapidIO AMI port to access additional statistics counters via the Application Register Interface (ARI) that are implemented in user logic, outside the IP core. These application-specific registers count occurrences of received packets, transmitted packets, errors, etc. These counts provide the user an idea of the amount of traffic being passed by the core, or can be used to verify packet transmission (i.e., send one NWrite and the count will increase by one).

As can be seen from the above description, the demo is primarily software driven. Packet types and transmission are controlled entirely by software through menu selections. While this demonstrates that the RapidIO 2.1 Serial Endpoint IP core is able to correctly handle the various RapidIO packet types, it does not demonstrate maximum throughput or provide a logical layer reference design. The demo validates correct operation in a Serial RapidIO system, which is achievable using the Soft Packet FIFO and the LatticeMico32.

Known Limitations

This section explains some of the limitations of the demo. These limitations are due primarily to the demo software not fully implementing all available features of the IP core or specification.

Fixed Link Speed

The link speed is fixed per bitstream settings. The speed cannot be changed dynamically. If a 4x 2.5Gbps Serial RapidIO demo bitstream is loaded, it can only be used to operate with other Serial RapidIO endpoints at this rate.

The link width is negotiable at run time. If operating with a 1x link partner, the IP core will resize the link to 1x, but the rate is still fixed at 2.5Gbps.

Only Small Transport Type Supported in Demo

While the IP core supports both large (16-bit ID) and small (8-bit ID) Transport Types (TT), the demo software does not fully handle the creating and parsing of RapidIO packets with 16-bit Transport Type fields. Therefore, only the smaller 8-bit ID is supported.

Only 34-Bit Addressing Supported in Demo

While the IP core supports 34-, 50-, and 66-bit addressing modes, the demo software only supports the creating and parsing of RapidIO packets with 34-bit addresses (the lower 32 bits are configurable, the upper two bits are fixed at 0). Regardless of address size selection in the menu, only 34-bit addresses will be generated.

Doorbell and PortWrite Not Supported in Demo

The demo software drivers and IP core support Doorbell and PortWrite type packets, but a menu page does not yet exist to exercise these packet transfer types. Future demo versions that operate with a provided third-party Serial RapidIO board will have doorbell demo capabilities.

TI 6472 Interoperability Not in Demo

The original demo plans called for including a Texas Instruments (TI) 6472 AMC board in the kit to show interoperability with a third-party Serial RapidIO device. Due to availability issues, we have only obtained one TI6472 AMC board, and so interoperability test development was not given high priority since we could not provide hardware to support this demo. Some menus still refer to TI6472 Interoperability, and these connection settings have been used in internal development for operating with TI DSPs, but the hardware is not provided in this demo.

Serial RapidIO Configuration Register Display May Not Support Third-Party AMC

The Register Display pages display the Serial RapidIO register set with version 2.1 extensions. When displaying configuration registers from a third-party version 1.2 or 1.3 board, certain bit fields may be interpreted incorrectly since they are reserved fields in the older Serial RapidIO version.

The Register Display pages are entirely correct for the RapidIO 2.1 Serial Endpoint IP core and mostly correct for the TI6472 DSP, but other third-party boards may have incorrect register displays.

Also, the register display does not track the extended features linked list. The locations of capabilities registers is hard-coded for the IP core or the TI6472 register sets. Unless other boards follow these register maps, the extended features may display incorrectly.

Set-up

This section describes how to connect and setup the LatticeECP3 AMC Serial RapidIO Demo.

General Warnings

Please observe the following important safety items:

Carefully Install the AMC Interface Cards

Please exercise caution when installing the AMC Interface Card, to ensure it is installed in the proper slot and oriented the correct way. The AMC connectors are not keyed, and can be installed in reverse which will short power and ground and other signals. For the purposes of this demo, the AMC Interface Card is only installed into the channel loopback slot.

Refer to the FTDI Communication Port Driver Installation section of this document for details on installing the LatticeECP3 AMC Evaluation Board into the AMC Interface Card. Additional information on board orientation and connector pinouts can be found in EB56, [LatticeECP3 AMC Evaluation Board - Revision B User's Guide](#).

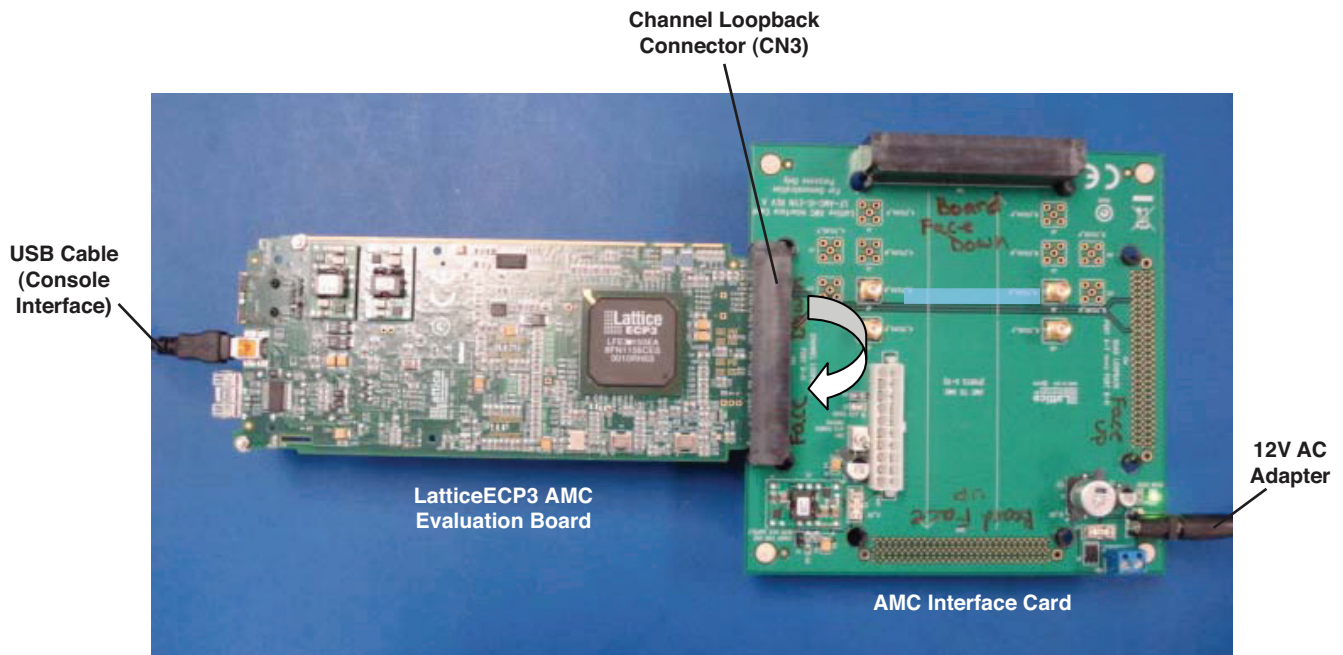
Remove Power First

Do not install or remove the AMC Interface Card without first disconnecting the power supply. If using the 12V AC Adapter power supply, unplug it from the AMC Interface Card power connector before installing or removing the card.

Demo Components

Figure 3 shows the LatticeECP3 AMC Evaluation Board installed in the AMC Interface Card channel loopback slot. All of these items are included with the LatticeECP3 Serial RapidIO Demo Kit.

Figure 3. LatticeECP3 AMC Serial RapidIO Loopback Demo Components



- **LatticeECP3 AMC Evaluation Board** – The evaluation board containing the RapidIO 2.1 Serial Endpoint IP core and the demo software. The demo bitstream is preloaded onto the LatticeECP3 AMC Evaluation Board. The only set-up and configuration necessary is to install the board into the Channel Loopback slot of the AMC Interface Card. See EB56, [LatticeECP3 AMC Evaluation Board - Revision B User's Guide](#) for complete details of the board.
- **AMC Interface Card** – Provides power and SERDES channel connectivity to the LatticeECP3 AMC Evaluation Board. The board is installed in the Channel Loopback slot so that it can talk to itself and not require any other AMC cards for operation. See Appendix A of EB56, [LatticeECP3 AMC Evaluation Board - Revision B User's Guide](#) for complete details on the AMC Interface Card.
- **12V AC Adapter** (not shown) – Provides the 12V power supply to the AMC Interface Card to power the LatticeECP3 AMC Evaluation Board.
- **Console Interface**
 - USB Cable – Standard Mini-USB cable that connects the AMC Interface Card console interface to the PC to display the menus.
 - Terminal Emulation Software – Provides connectivity to the PC Comm port and display for the menus. HyperTerminal or other serial-port based VT100 emulation software can be used.

Note: Windows Vista and Windows 7 do not include HyperTerminal. Users must install a suitable serial port terminal emulator such as PuTTY 0.60 or the Windows XP version of HyperTerminal. The word "HyperTerm" or "HyperTerminal" may still be used throughout this document to indicate a VT100 terminal emulation program, but the

preferred choice is to use PuTTY. All menu examples show a PuTTY terminal and the setup instructions are for PuTTY.

Evaluation Board Installation

The AMC connectors on the AMC Interface Card are not keyed. It is important to install AMC Interface Cards in the proper orientation or power may be shorted to ground. The AMC Interface Card has four AMC connectors. An AMC Interface Card is installed face up in the two connectors on the underside of the board (CN2: AMC to AMC, and CN4: Quad Loopback). An AMC Interface Card is installed face down in CN1: AMC to AMC and CN3: Channel Loopback.

Figure 3 shows the LatticeECP3 AMC Evaluation Board installed in the Channel Loopback slot. Note that the AMC is face down in the Channel Loopback connector. The LatticeECP3 device is visible because it is on the back side of the board.

Refer to EB56, [LatticeECP3 AMC Evaluation Board - Revision B User's](#) for an illustration of the orientation of the LatticeECP3 AMC Evaluation Boards installed into each slot.

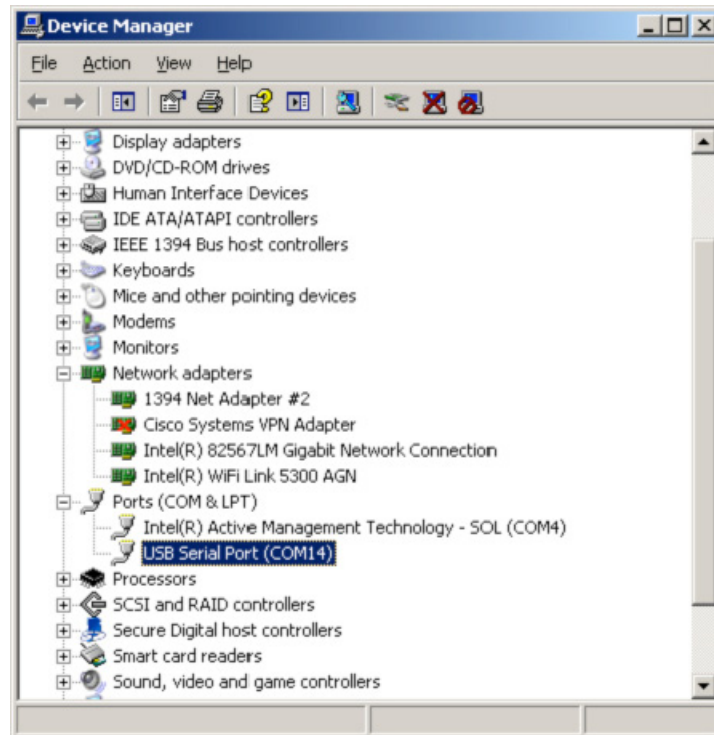
FTDI COM Port Driver Installation

The next step is to connect the USB cable to the mini-USB connector on the front of the LatticeECP3 AMC Evaluation Board. Connect the other end to an available USB slot on your PC. This will provide the connection to the menus on the board. You will need to install a driver provided by Future Technology Devices, International (FTDI), the manufacturer of the FT232RL device used on the LatticeECP3 AMC Evaluation Board that converts the RS-232 UART output to USB. The driver converts the PC USB port to a comm port so that a serial terminal program (HyperTerminal) can open the USB port and communicate over it as if it were an RS-232 port. The FTDI drivers can be downloaded from the FTDI web site.

Note: Windows Vista and Windows 7 do not include HyperTerminal. Users must install a suitable serial port terminal emulator like PuTTY 0.60 or the Windows XP version of HyperTerminal.

Once the driver is installed, power up the LatticeECP3 AMC Evaluation Board by connecting the power supply.

To determine which comm port the USB serial device is mapped to, open **My Computer > Properties > Hardware-> Device Manager** and select **Ports** to see the available serial ports and choose the one that is the USB Serial Port. See Figure 4 as an example.

Figure 4. Device Manager Dialog Box

Open HyperTerminal or PuTTY and select the new USB Serial Port. Set the communication settings to: **115000 baud, 8 data bits, No Parity, 1 stop bit (115000,8,N,1)**. Figures 5 and 6 show these settings for a PuTTY session.

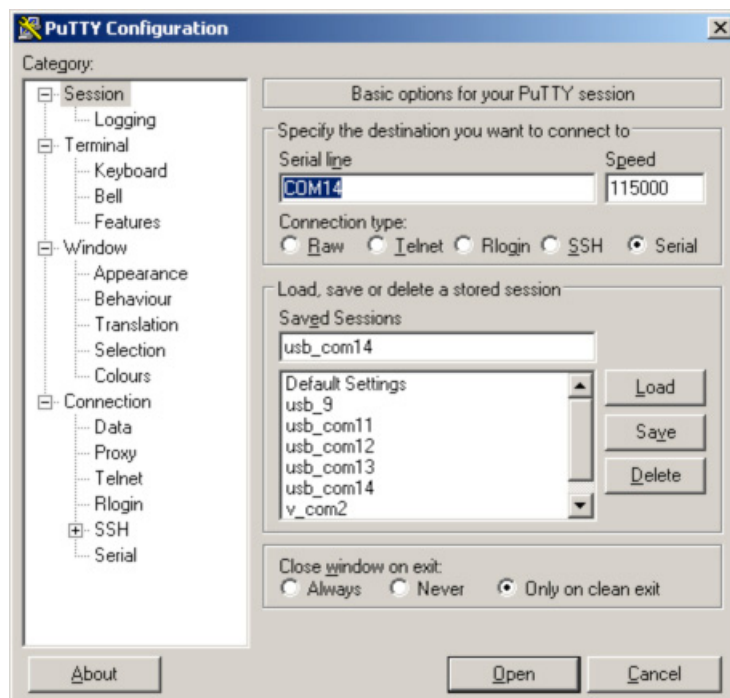
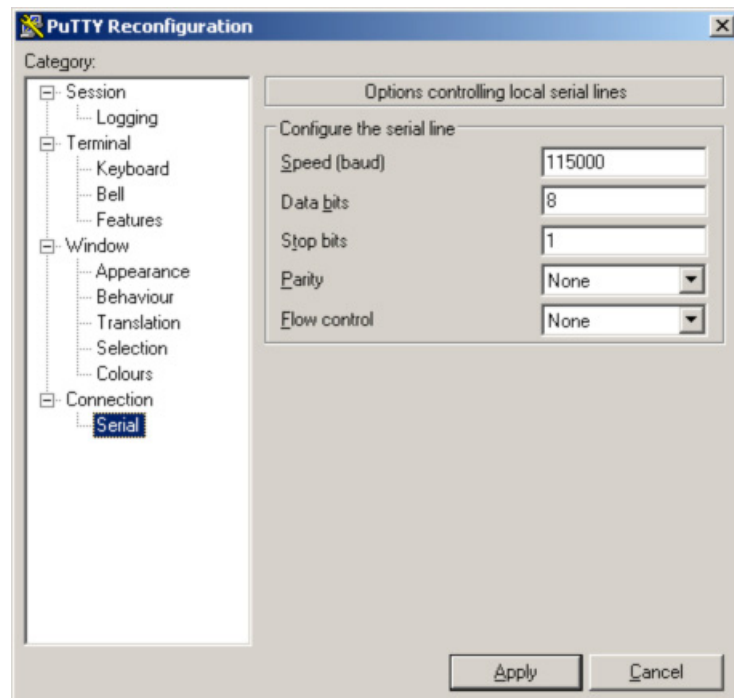
Figure 5. Set the Communication Settings

Figure 6. Set the Communication Settings

After the serial port settings are configured, and power is applied to the board, press **Enter** to redisplay the main menu.

Hardware Settings

For the default loopback demo operation, there is no hardware configuration necessary. This section is provided strictly for reference, or in case the LatticeECP3 AMC Evaluation Board or AMC Interface Card has been used in other applications and their current configuration is in doubt.

LatticeECP3 AMC Evaluation Board

There are no jumpers or switches that need to be set to configure the demo, or during demo operation. All jumpers are preset at the factory for correct clocking and JTAG scan chain operation.

For reference purposes only:

- J2 – Removed
- J11 – Removed
- J13 – Removed
- J14 – Install on left pair
- J15 – Install on 2-3
- SW3 – Don't care

The only hardware indicator of demo activity is the blinking red LED D25 that indicates the system clock is running, the bitstream has loaded correctly and basic logic is functional. It does not reflect the Serial RapidIO or LatticeMico32 operating status.

No other LED indicators are used.

AMC Interface Card

For normal demo operation, no changes need to be made to the AMC Interface Card. The only jumper on the AMC Interface Card provides a selection for using an ATX power supply versus the 12V AC Adapter. The ATX power supply is used in situations where multiple, high-power evaluation boards are installed into the AMC Interface Card for interoperability. For the simple loopback demo using a single AMC Interface Card, the 12V AC adapter power supply is adequate.

J2 – removed for 12V AC adapter usage, installed for ATX power supply usage.

Demo Menus

A serial port running over a USB cable to a PC provides the user interface for running the demos on the LatticeECP3 AMC Evaluation Board. All user interaction and demo control is handled through the menus. This section discusses the demo pages that define the features and functionality of this demo.

General User Interface

The menus are text-based, using the VT100 terminal commands to draw text at specific locations and with specific attributes. This is a more user-friendly menu system than most command line interfaces. This section provides an overview of the user interface operations required to navigate the menus and operate the demo.

Single-Key Entry

For most operations, only a single letter or number key needs to be pressed. You do not need to enter the letter/number and then press Enter. This greatly speeds the menu navigation process by eliminating the extra Enter step.

Some pages require a key to be pressed to select an operation. Once selected, other user entry methods may be used to edit the fields.

Select Fields

When a limited number of selections are permitted for a field, a Select Field is used. This is similar to a list box in a GUI. Usually a key is pressed to activate the Select Field, such as selecting the Serial RapidIO packet type. The text area is highlighted to show that the field is activated. Press the **Space** bar to toggle through the selections for the field. In the example above, the **Space** bar will cycle through the selections: MaintRd, MaintWr, NRead, NWrite, NWrite_R, and SWrite. Once the desired selection is displayed, press the **Enter** key to select the entry. Press the **Esc** key to abort the choice and restore the original setting.

Hex Numbers Fields

Certain fields require a hexadecimal value to be entered, such as addresses, data values, and IDs. When entering a hexadecimal value, the field will accept the keys 0-9 and A-F. Do not enter a prefix such as "0x" or "32'h".

Leading 0's are acceptable, but not necessary. Field entry size may be limited to the size of the data expected, such as two characters for a byte, four characters for a short (16 bits) and eight characters for a word (32 bits).

Decimal Number Fields

Certain fields require a decimal value to be entered, such as length. When entering a decimal value, the field will accept the keys 0-9. Do not enter negative or positive signs.

Leading 0's are acceptable, but not necessary. Field entry size may be limited to the size of the data expected, such as three characters for an expected value in the range of 0-100.

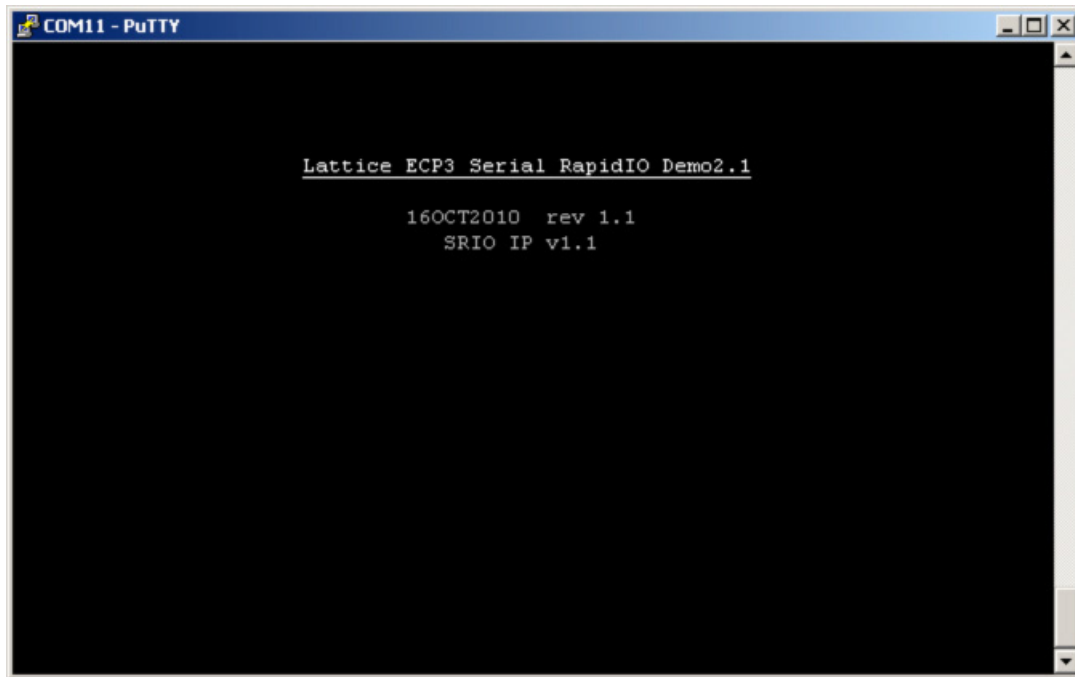
Navigation

Some pages require movement to different lines for configuration. The up and down arrows are used to move to a new page or row.

About Page

The About page is displayed upon power-up or reset. It clears in approximately three seconds and moves to the Main Menu page. The purpose of the About page is to display the demo version number information. From the Main Menu, the About page can be displayed by pressing the **A** key. This allows the user to view the version information while the demo is running. Press **Enter** to return the Main Menu.

Figure 7. About Page



Main Menu Page

The Main Menu page is the starting point for accessing all the other pages in the demo software. Simply press the highlighted letter to go to the corresponding page.

The link state is shown in the lower-left corner as a quick indicator of link status. Upon entering the Main Menu, check that the link state is connected as either a 1x, 2x or 4x. More details on the link state can be found on the Status page.

Figure 8. Main Menu Page

- Press **C** to display the **Demo Configuration** page
- Press **S** to display the **SRIO IP Status** page
- Press **R** to display the **SRIO Register Display** page
- Press **P** to display the **Packet Transfer** page (to send individual packets)
- Press **I** to display the **InterOp Tests** page to run pre-defined tests that exercise Maintenance and Memory transactions in a PASS/FAIL test scenario
- Press **V** to display the **View Local Memory** page to examine the target memory that is read/written by the IP core from the CPU side to verify that changes have taken place during Serial RapidIO transfers

Pressing **Enter** causes the page to be refreshed. This is helpful if you connect to a board in which the demo is already running and you wish to bring up the Main Menu again. Otherwise, the terminal window will not be updated and will remain blank.

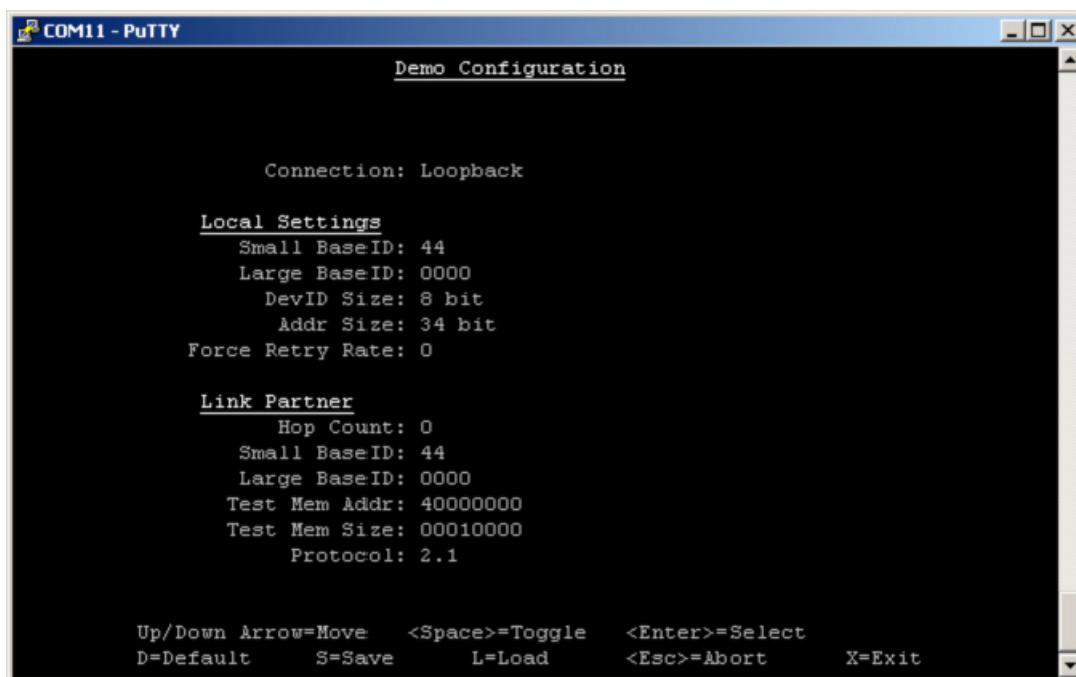
Demo Configuration Page

This page allows you to set certain parameters used in communicating with Serial RapidIO endpoints over the Serial RapidIO fabric. The default Loopback Connection settings are the standard settings. If you would like to configure the AMC Interface Card to talk with another card, settings such as the BaseIDs, Test Memory Address and Test Memory Size must be changed so the tests can access the other card.

No changes to this page are necessary to use the demo in the standard loopback configuration. These are the defaults for operating the demo looped back to itself.

To change a line, navigate to the line to be changed using the up and down arrow keys. Then press **Enter** to edit the field.

Figure 9. Demo Configuration Page



Settings

Connection: The Connection field is a Select type field. Use the arrow keys to navigate to this line and press **Enter**. Once selected (highlighted), press the **Space** bar to toggle through the available selections.

- Loopback – Standard demo set-up for talking to itself.
- LatticeECP3 AMC Evaluation Board – Settings to communicate with a second evaluation board if installed in the AMC-to-AMC slots on the AMC Interface Card.
- TI6472 – Settings used to operate with a third-party Serial RapidIO board.
- Custom – User-configurable settings that will not be overridden by defaults (see below).

Local Small BaseID: Hexadecimal entry number field for selecting the Serial RapidIO 8-bit BaseID value used in the Serial RapidIO packet source ID field. Press **Enter** to select the field and then enter the hexadecimal value. Press **Enter** to accept the value.

Local Large BaseID: Configures the large (16-bit) base ID value used in the large TT packet format. Note that while this field can be set, the software does not support the large TT packet format, so it is not used.

DevID Size: Select this field to determine whether 8-bit or 16-bit TT format is used (only 8-bit is available at this time).

Addr Size: Use this field to select either 34-bit, 50-bit or 66-bit addressing for RapidIO packets. Although selectable, only 34-bit addressing is supported by the software, so this field value does not affect packet generation.

Force Retry Rate: Decimal value to select how often to generate a retry on a received packet. This is currently not implemented, but will be used in the future with the Throughput demo to test back-pressure and flow control effects on packet throughput.

Hop Count: Decimal number entry field to specify the number of hops (in Maintenance packets) to reach the link partner. This setting is used to send Maintenance packets through unconfigured Serial RapidIO switches to end-points. This field is inserted into the Maintenance packet hop count field.

Link Partner Small BaseID: Hexadecimal entry number field to select the 8-bit BaseID value used in the Serial RapidIO packet destination ID field. Press **Enter** to select the field and then enter the hexadecimal value. Press **Enter** to accept the value.

Link Partner Large BaseID: Configures the large (16-bit) base ID value used in the large TT packet format. Note that while this field can be set, the software does not support the large TT packet format, so it is not used.

Test Mem Addr: A hexadecimal field entry that specifies the address of the link partner to use for memory read/write tests. The default value for AMC loopback is 0x40000000 which will access the Serial RapidIO test memory via the Logical Port #1 interface. For other interoperability situations, ensure that you know which memory on the link partner board can be accessed without adverse effects.

Test Mem Size: A hexadecimal field entry that specifies the size of the link partner's test memory for read/write tests. The default value for AMC loopback is 64kB which is the size of the Serial RapidIO test memory. For other interoperability situations, ensure you know the memory size on the link partner board.

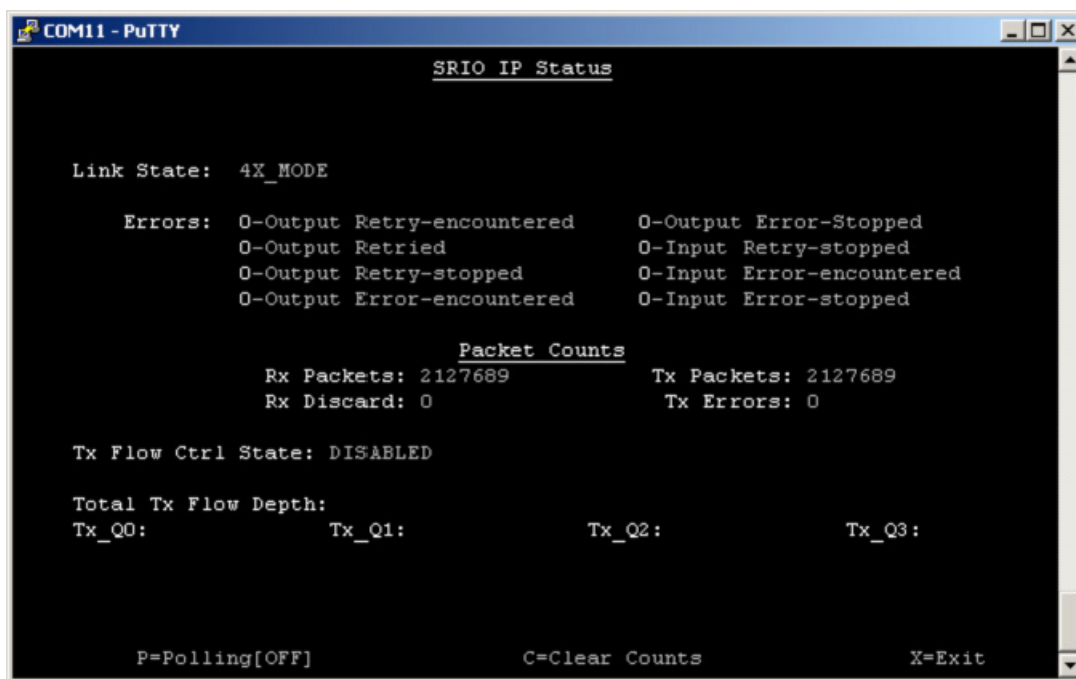
Saving Settings: If you want to return a setting to its default state, first set the connection type (i.e. loopback), press **Enter** to select it and then press **D** to reload the default settings for that connection type.

The demo software allows customized changes to be saved in EEPROM for later use. Press the **S** key to save all the demo settings to EEPROM memory. To recall the settings, press the **L** key to load them from EEPROM. If a valid set of settings has not been previously saved to EEPROM, an error beep will be produced and no values will be changed.

SRIO IP Status Page

The SRIO IP Status page displays the link state and packet and error counts. The link state information is obtained from the IP core link state machine.

Figure 10. SRIO IP Status Page



Link states:

- **SILENT** – Pause time at the beginning of the state machine
- **SEEK** – No symbols are detected on any channels (card may not be installed in the loopback slot)
- **DISCOVERY** – Attempting to determine the number of lanes to include in a link
- **1X_MODE_LANE0** – Linked as 1x on channel 0
- **1X_MODE_LANE1** – Linked as 1x on channel 1
- **1X_MODE_LANE2** – Linked as 1x on channel 2
- **1X_RECOVERY** – Link retraining due to numerous errors or loss of symbol lock
- **2X_MODE** – Linked as 2x on channels 0 and 1
- **2X_RECOVERY** – Link retraining due to numerous errors or loss of symbol lock
- **4X_MODE** – Linked as 4x on using all channels

Packet counts are obtained from the Serial RapidIO status vector and converted to numeric values in user logic and read via the ARI.

See the Link Status Interface section in IPUG84, [RapidIO 2.1 Serial Endpoint IP Core User's Guide](#) for more details on the link state vector and the packet and error counts.

Tx Flow Control is not enabled in the RapidIO 2.1 Serial Endpoint IP core and therefore shows as disabled and no counts are provided.

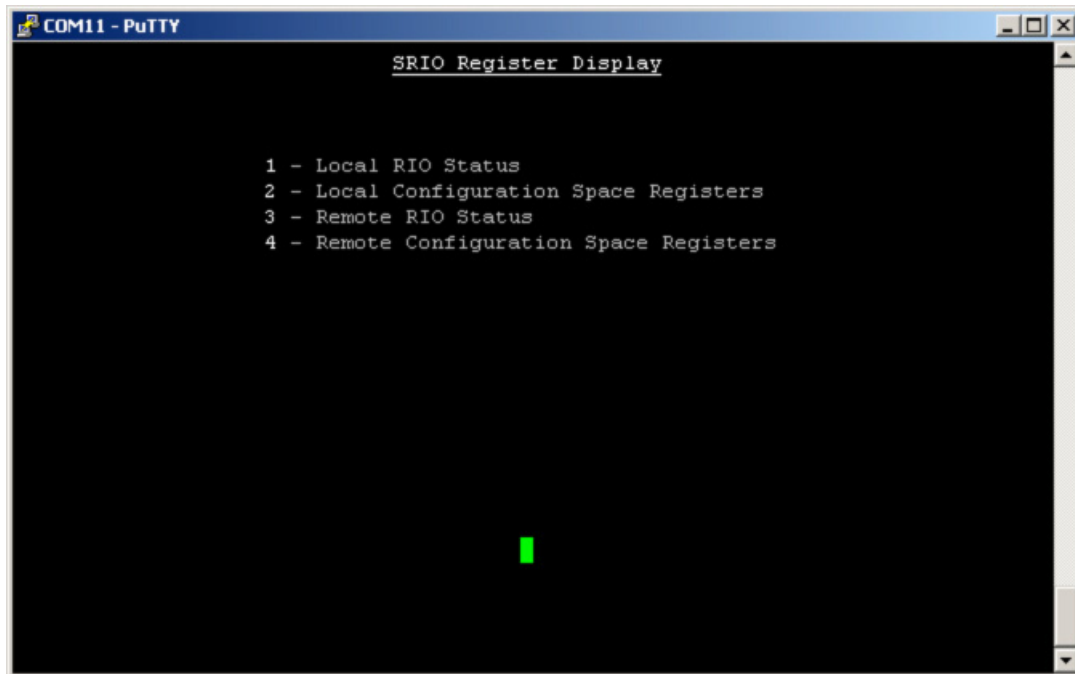
Press **C** to clear the packet and error counters.

Press **P** to enable automatic polling for new packet counts and errors. This is not applicable in a loopback configuration because you need to be on another page in order to generate traffic. This is useful in an interoperability situation where the link partner is sending and this is automatically displaying the new counts.

SRIO Register Display Pages

The SRIO Register Display pages show the Serial RapidIO configuration and status registers. This provides a means to verify access to a link partner's configuration registers via Maintenance transactions. It also can be used to inspect the status of the link and its capabilities.

The SRIO Register Display pages are organized into two categories: local registers (reading the Serial RapidIO IP registers directly through the AMI port) and remote link-partner registers (accessed via Serial RapidIO Maintenance Reads). In each category, the RapidIO Status option decodes certain status register bit fields into a more readable format. The other option moves all registers into hexadecimal format.

Figure 11. SRIO Register Display Page

The SRIO Register Display pages allow examining such registers as the COMP_TAG register and the BASE_DEV_ID register which are affected by running various demo tests.

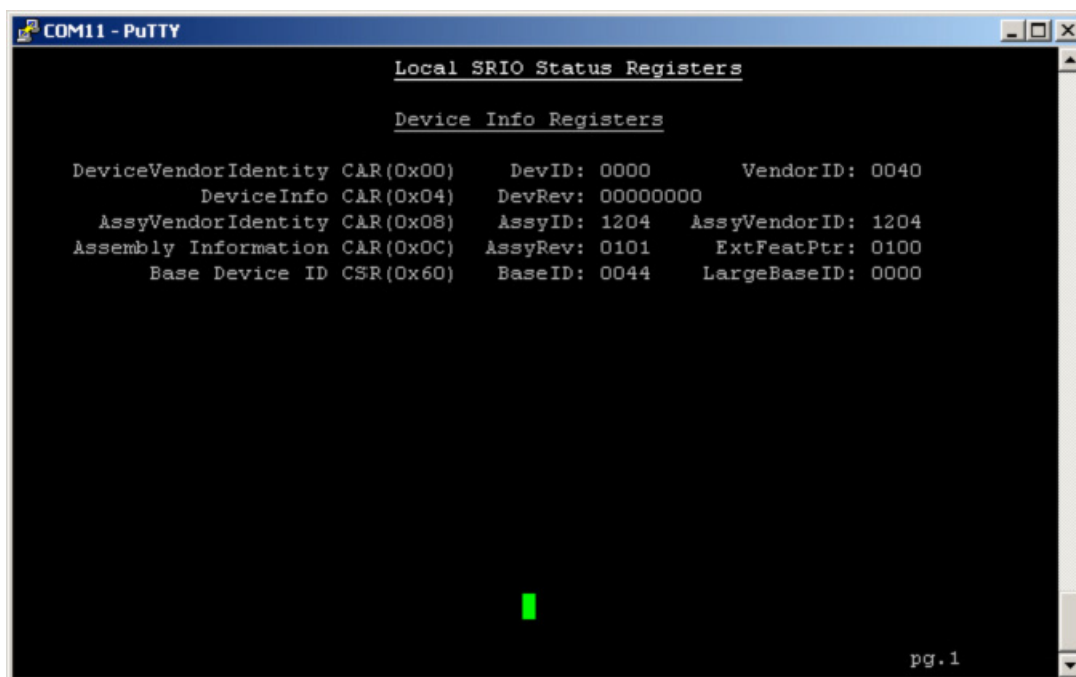
The COMP_TAG is used as a scratchpad to write new values and then read them back and verify them in the InterOp Tests page.

BASE_DEV_ID is changed via the Demo Configuration page and shows that Maintenance writes to the core have had an effect.

Figure 12 is an example of the first page of the Local SRIO Status Registers pages. Use the up and down arrow keys to scroll through the various pages.

Esc or **X** exits back to the main SRIO Register Display page.

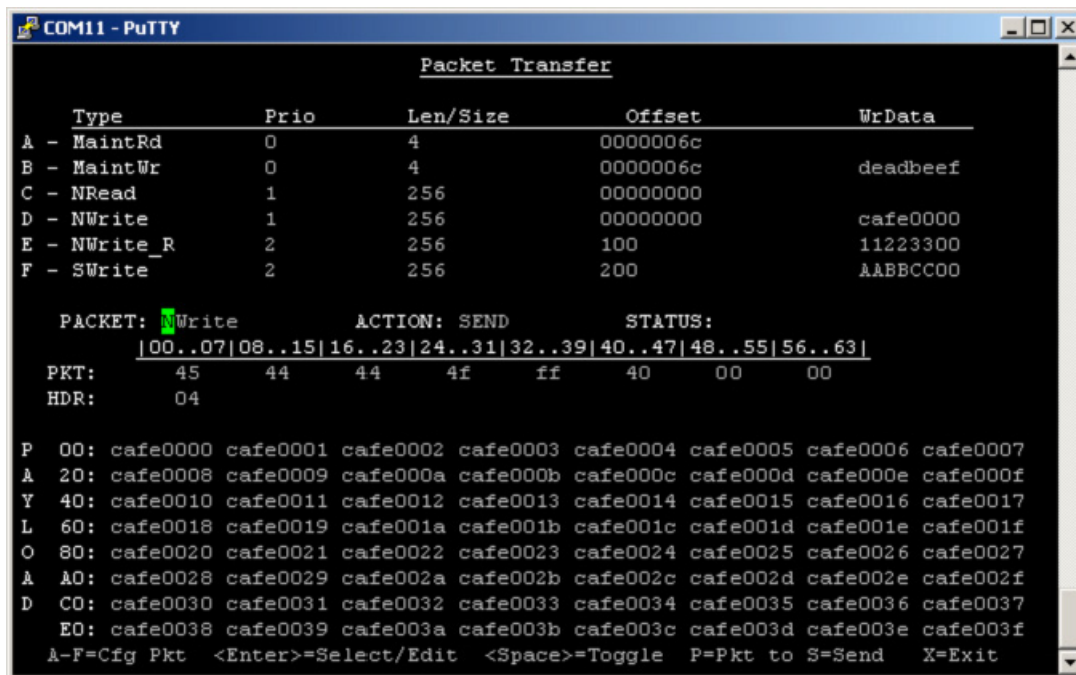
Figure 12. First Page of the Local SRIO Status Registers Page



Packet Transfer Page

The Packet Transfer page allows configuration of various supported Serial RapidIO packet types. Once configured, the packet to be sent is selected, and can then be sent over the Serial RapidIO link. The status of the operations is reported, along with any returned payload.

Figure 13. Packet Transfer Page



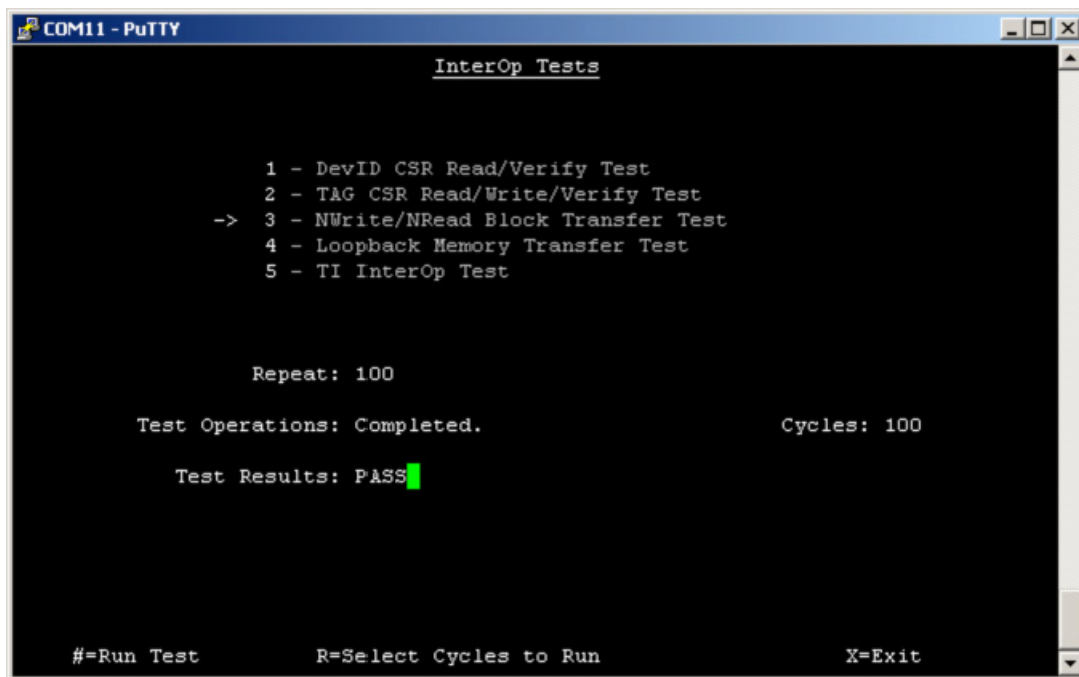
To configure and select a packet to be sent:

- Press **A** through **F** to select the packet type to configure
- Select the priority field value the using the **Space** bar to toggle through values 0-3.
- Len/Size is a toggle selection to select the size of the transfer to common values. The drivers do not support the formatting of abnormal-sized transfers (e.g., 73 bytes).
- Offset is a hexadecimal entry field for specifying the address to read/write to. This value is added to the Test Memory Address set in the Demo Configuration Page and is used for Maintenance transactions.
- WrData is a hexadecimal entry field for write transactions. If the size of the transfer is greater than 4 bytes, the following words will have an incrementing pattern.
 - Press **P** to choose the packet type to send. This brings up the selection list. Use the **Space** bar to toggle through the list and select the packet to be sent.
 - Press **S** to send the packet type chosen with **P**. The packet is sent and if a response is expected, it waits and displays the return status and payload.

InterOp Tests Page

The InterOp Tests Page allows pre-set tests to be run to verify that Maintenance and Memory packet operations are error-free over long test periods.

Figure 14. InterOp Tests Page



Tests

- **Test #1** – Reads the DevID CSR from the link partner. The returned read values are compared with each other over “Repeat” number of times. If the read values do not match between successive reads the test aborts with an error. This tests access to the link partner’s configuration registers.
- **Test #2** – Writes and reads the CSR Tag register. The CSR Tag register is a 32-bit scratchpad register. The written value is incremented each cycle. This test verifies both the maintenance write and read accesses to the link partner.
- **Test #3** – Performs memory NWrite transfer to the Link Partner test Memory for Test Memory Size (see the

Demo Configuration page for these settings). The test then reads back the far-end memory and to verify that the correct values were read back.

- **Test #4** – Similar to test #3, but the memory contents are also verified by the CPU access to the test memory. The test memory contents are inverted for the Serial RapidIO read-back portion of the test to confirm that the changed data is seen at the Serial RapidIO SPF. This test is only available in loopback connection mode since the CPU needs to access the test memory.
- **Test #5** – Not implemented.

Controls

Repeat: Press **R** to select the Repeat field. This field sets the number of times to run a specific test. Use the **Space** bar to toggle through the list and press **Enter** to select. The option **FOREVER** allows continuous running. Once a test sequence is running, pressing any key will abort the test cycle.

Test Selection: Press the number of the test to run it. An arrow will indicate the running test. This example shows that Test #3 was selected to run 100 times and that it has completed successfully.

Cycles: Displays the current cycle count of the running test. This field is automatically updated as the test runs. The field increments by 10,000 for Tests #1 and #2 since they are simple, single-register accesses. The field increments by 2 for Tests #3 and #4 since they are lengthy tests involved with reading and writing 64kB per test cycle and take much longer to complete.

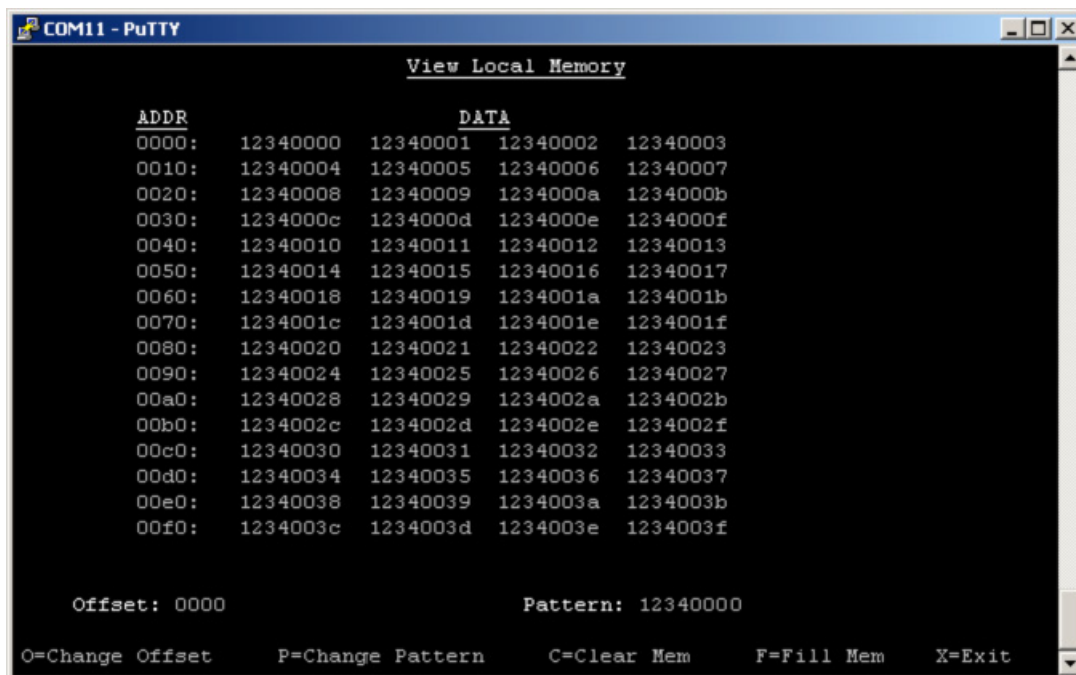
Test Operations: Displays the operations the test is performing, and shows “Completed” when the test finishes the number of cycles successfully.

Test Results: Displays “PASS” when the test completes successfully or “FAIL” if an error is detected at any time during the test.

View Local Memory Page

The View Local Memory page allows the user to view the Serial RapidIO test memory from the CPU/software side to verify that SRIO NWrite operations have written the correct data into the destination memory. The user can also set a pattern into the memory to be read back later using the SRIO NRead operation.

Figure 15. View Local Memory Page

**Controls**

Offset: Press **O** to select the Offset field and enter an address to display. For example, to see the last 256 bytes of the test memory, enter **0xff00**. The range must be within 64kB.

The up and down arrow keys can be used to scroll from page to page. The Offset entry is used to make large jumps to other sections of the memory.

Pattern: Press **P** to select the Pattern field and enter a 32-bit hexadecimal pattern to start filling the memory. The value entered will be written to offset 0, then pattern+1, pattern+2, etc. is written to each subsequent word.

Fill Mem: Press **F** to fill the memory with the 32-bit incrementing hexadecimal pattern.

Clear Mem: Press **C** to clear the entire memory to all 0's.

Up Arrow/Down Arrow: Use the up and down arrow keys to scroll to different memory pages.

Demo Scenarios

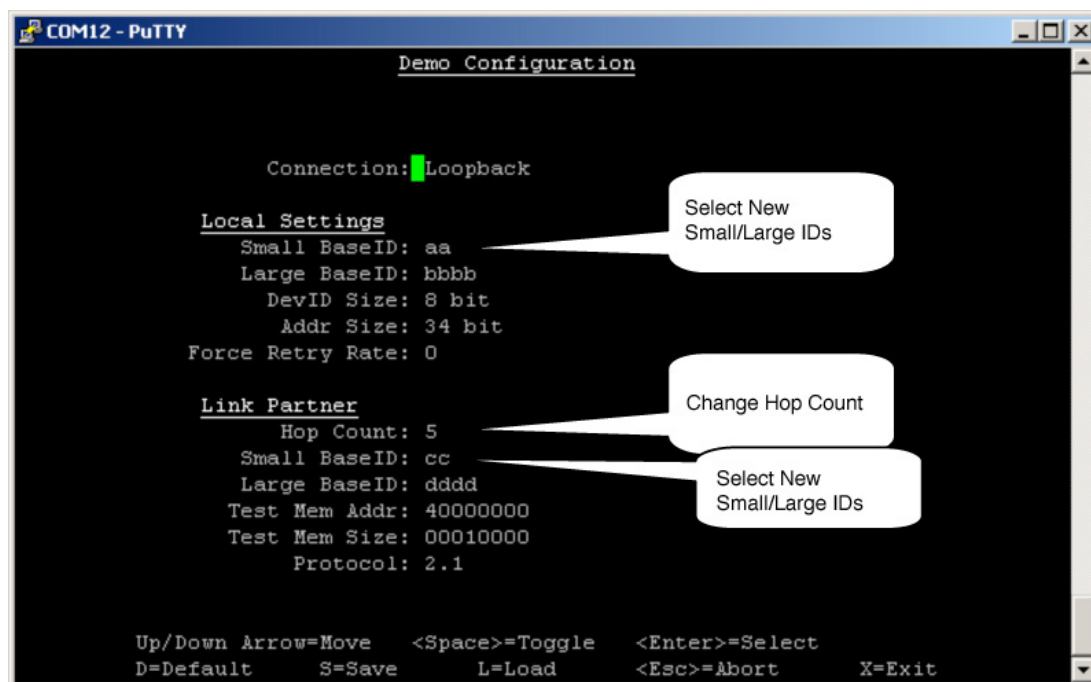
This section provides step-by-step instructions for executing various demos. This will include information on demo page operation and how various pages can be used together to initiate Serial RapidIO actions and check the results.

Modifying Configuration Settings and Verifying

This sequence shows how to change various Serial RapidIO link parameter settings and view the resulting changes to the CSR registers and RapidIO packet header.

From the main menu, press the **C** key to enter the Demo Configuration page. We will change the Local Small BaseID and Large BaseID settings, the Hop Count and the destination Link Partner Small BaseID and Large BaseID to see how the changes affect RapidIO packet header generation.

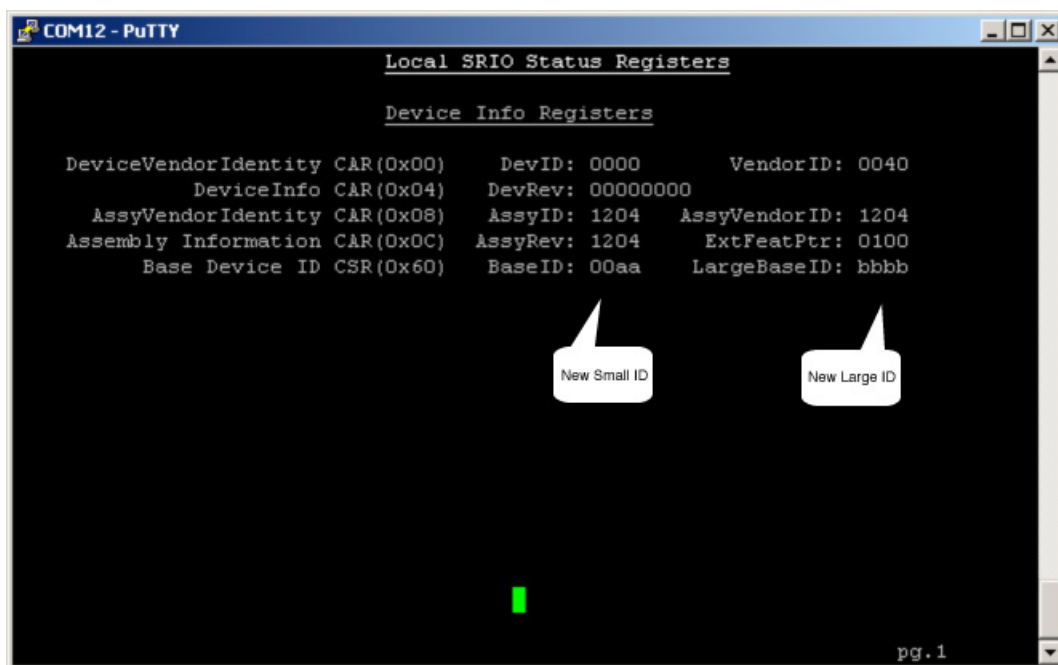
Figure 16. Modifying the Configuration Settings



1. Use the **down arrow** key to move the cursor to the Small BaseID row. Press **Enter** to edit. Enter the new hexadecimal value **AA**. Press **Enter** to save the change.
2. Use the **down arrow** key to move the cursor to the Large BaseID row. Press **Enter** to edit. Enter the new hexadecimal value **BBBB**. Press **Enter** to save the change.
3. Use the **down arrow** key to move the cursor to the Hop Count row. Press **Enter** to edit. Enter the new decimal value **5**. Press **Enter** to save the change.
4. Use the **down arrow** key to move the cursor to the Link Partner Small BaseID row. Press **Enter** to edit. Enter the new hex value **CC**. Press **Enter** to save the change.
5. Use the **down arrow** key to move the cursor to the Link Partner Large BaseID row. Press **Enter** to edit. Enter the new hex value **DDDD**. Press **Enter** to save the change.

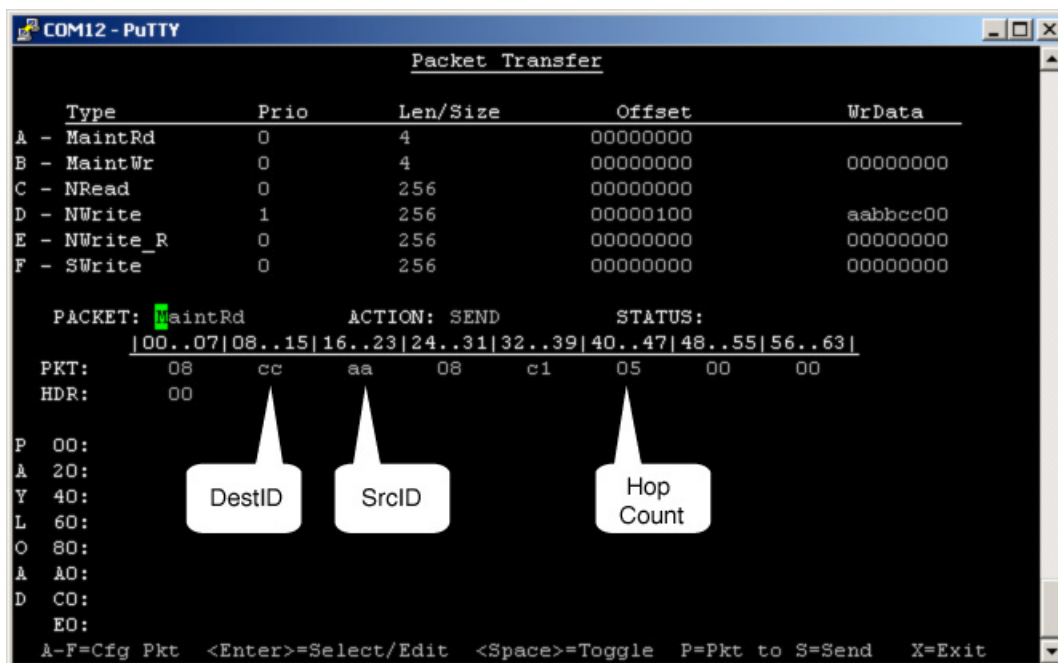
Return to the Main Menu page by pressing **X**. Then press the **R** key to enter the SRIO Register Display page, and select **1** to display the Local SRIO IP register settings. Figure 17 shows that the small and large Base IDs have been updated with the changes made in the Demo Configuration page.

Figure 17. Verifying the Configuration Setting Changes



Exit back to the Main Menu by pressing X twice. Then enter the Packet Transfer page and select a MaintRd to be sent (this will be the default selection). Note that the RapidIO packet header fields now contain the updated values from the Demo Configuration page. Using the Demo Configuration page, the user can change the Source, Destination, Hop Count and Base Memory Address fields of the packet. Large Transport Type (TT) is not supported by the software, so the 16-bit BaseIDs are not used in packet header generation.

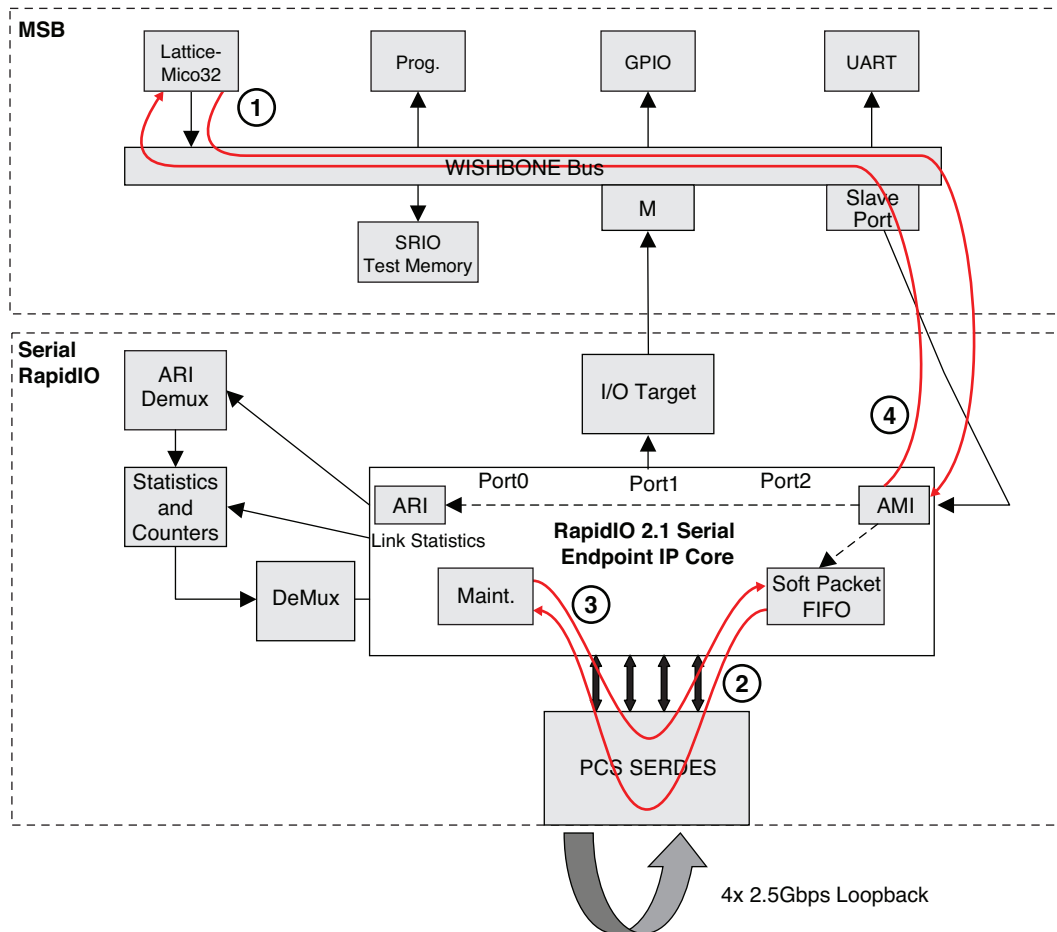
Figure 18. Verifying the Configuration Setting Changes



Demonstrating Maintenance Reads/Writes to TAG CSR

Figure 19 illustrates the steps and data flow of the Maintenance Read Packet Transfer test. A single Maintenance Read packet is sent using the Packet Transfer page. In this example, the Maintenance Read packet is configured to access the TAG CSR.

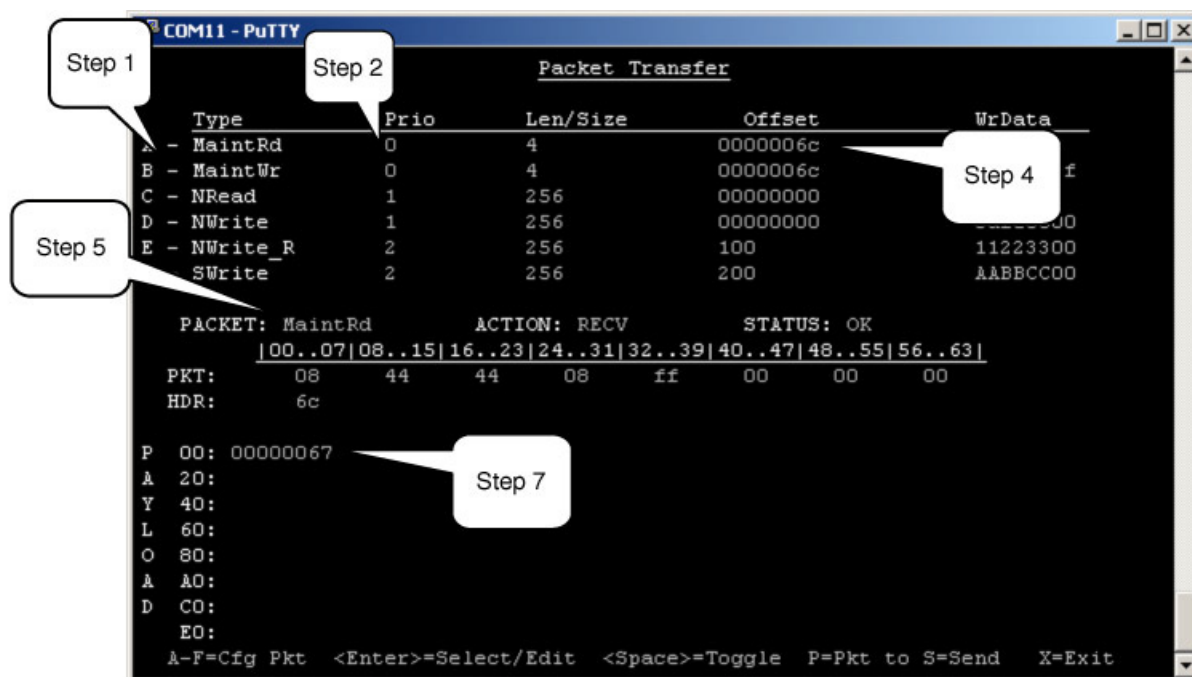
Figure 19. Maintenance Read Packet Transfer Test



1. Software running on the LatticeMico32 CPU creates a Maintenance Read RapidIO packet and loads it into the SRIO Soft Packet FIFO (SPF) via register writes to the AMI port. The AMI port is accessed through a slave port component (SP) on the LatticeMico32 WISHBONE bus.
2. The SPF then transfers the RapidIO packet into the transport layer and out the SERDES. The SERDES are looped back by the channel loopback connector, and so the packets are received by the RapidIO 2.1 Serial Endpoint IP core and forwarded to the internal Maintenance/Management port by the DeMux Policy. The addressed TAG CSR is then accessed and the current value obtained.
3. A Maintenance Response packet, with the obtained TAG register value, is created and submitted to the transport and PHY layer for transmission onto the Serial RapidIO link. The packet is sent and arrives back at the SPF.
4. When the SPF has received the response, the software reads the response packet from the SPF and decodes the header and displays the payload.

Figure 20 shows how to set up the transfer.

Figure 20. Packet Transfer Settings for Maintenance Read Packet Transfer Test



1. Press **A** to edit the MaintRd packet configuration line.
2. Use the **Space** bar to toggle and select the priority which the Maintenance Read should be sent. Press **Enter** to select.
3. Len is fixed at 4 bytes by the demo, so this field is skipped automatically.
4. The Offset field is now editable. Enter **6C** (hexadecimal) for the register address to be read. This is the TAG register.
5. Press **P** to select the SRIO packet to be sent. Use the **Space** bar to toggle through the selections until MaintRd is displayed and press **Enter** to select. The packet header is now displayed.
6. Press **S** to send the selected packet (MaintRd).
7. The packet is sent via the Soft Packet FIFO, and the received response payload is displayed in the payload section, along with the transfer status ("OK" for successful transmission and reception).

As a companion exercise, select the MaintWr packet configuration and change the value to be written to **TAG CSR**.

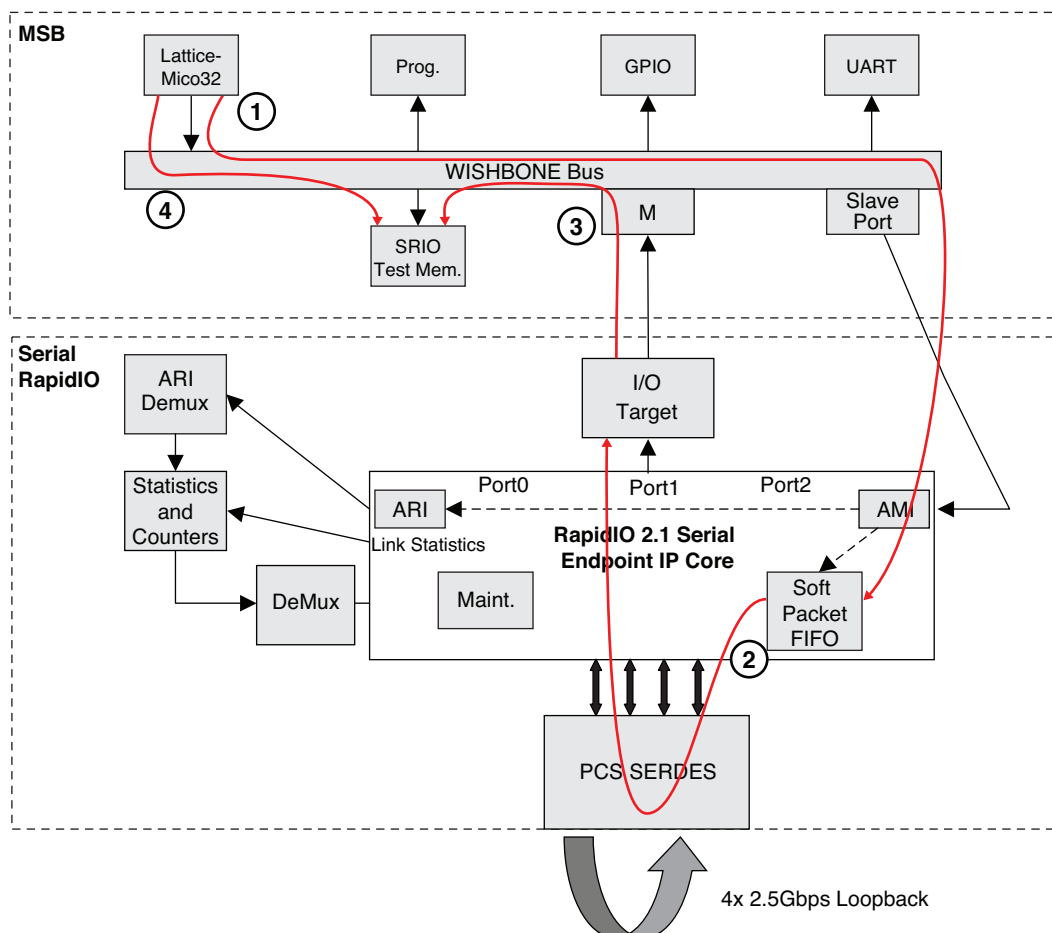
1. Press **C** to edit the MaintWr packet configuration line.
2. Use the **Space** bar to toggle and select the priority at which the Maintenance Write should be sent. Press **Enter** to select.
3. Len is fixed at 4 bytes by the demo, so this field is skipped automatically.
4. The Offset field is now editable. Enter **6C** (hexadecimal) for the register address to be read. This is the TAG register.
5. The WrData field is now editable. Enter a unique hexadecimal number to be written to the TAG register.

6. Press **P** to select the SRIO packet to be sent. Use the **Space** bar to toggle through the selections until MaintWr is displayed and press **Enter** to select. The packet header is now displayed.
7. Press **S** to send the selected packet (MaintWr).
8. The packet is sent via the Soft Packet FIFO, and the value is written into the CSR TAG register.
9. Press **P** to select the packet type to be sent, and toggle through until MaintRd is displayed. Press **Enter**.
10. Press **S** to send the packet. As in the previous demo, the value read from the TAG register will be displayed in the Payload field.
11. Return to the Main Menu page and press **R** to enter the SRIO Register Display page. Press **2** to enter the Local SRIO Registers page and note the value displayed for the TAG register (0x6C). This should be the value written in the previous steps.

Demonstrating NWrite Memory Transfers

The following steps illustrate the configuration of an NWrite Packet and the subsequent transfer of the packet to test memory using the Packet Transfer page. The flow of data is very similar to the previous description of sending the Maintenance Read packet. Figure 21 illustrates the data flow.

Figure 21. NWrite Memory Transfer Data Flow

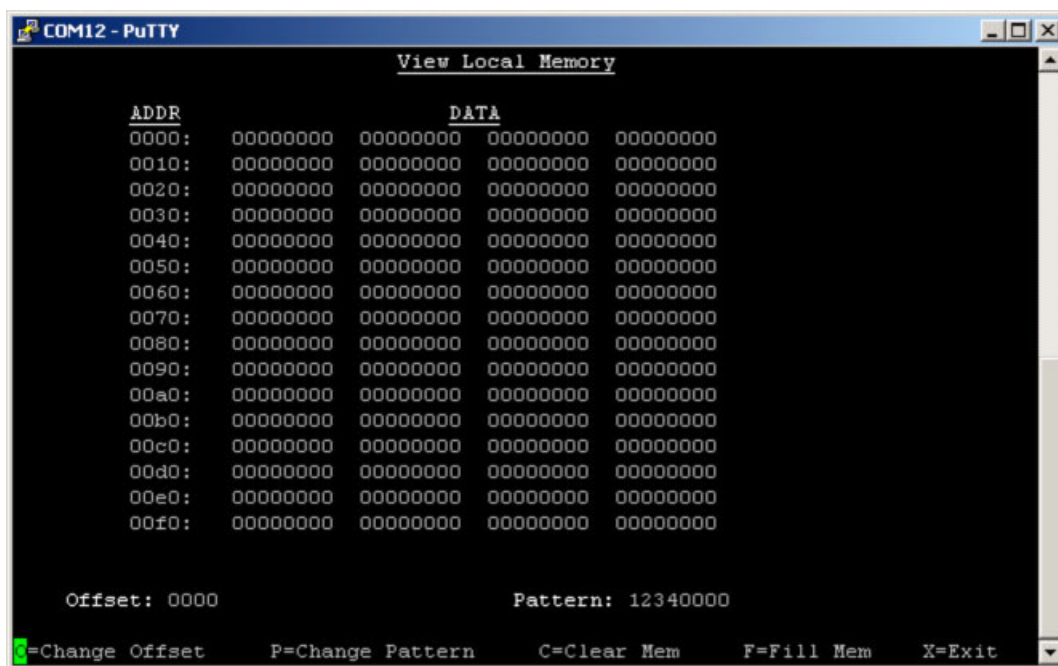


1. Software running on the LatticeMico32 CPU creates an NWrite RapidIO packet with the appropriate size payload and loads it into the Soft Packet FIFO (SPF) via register writes to the AMI port.

2. The SPF then transfers the RapidIO packet into the transport layer and out the SERDES. The SERDES are looped back by the channel loopback connector. The packet is received by the IP core and forwarded to Logical Port #1 port by the DeMux Policy.
3. The I/O Target module receives the RapidIO Memory packet from the IP core and converts it into a WISH-BONE burst write. The burst write is targeted to the 64kB SRIO Test Memory component in the LatticeMico32 platform.
4. After the packet transfer is complete, the software inspects the test memory contents to verify that the data pattern has been written into the correct memory locations.

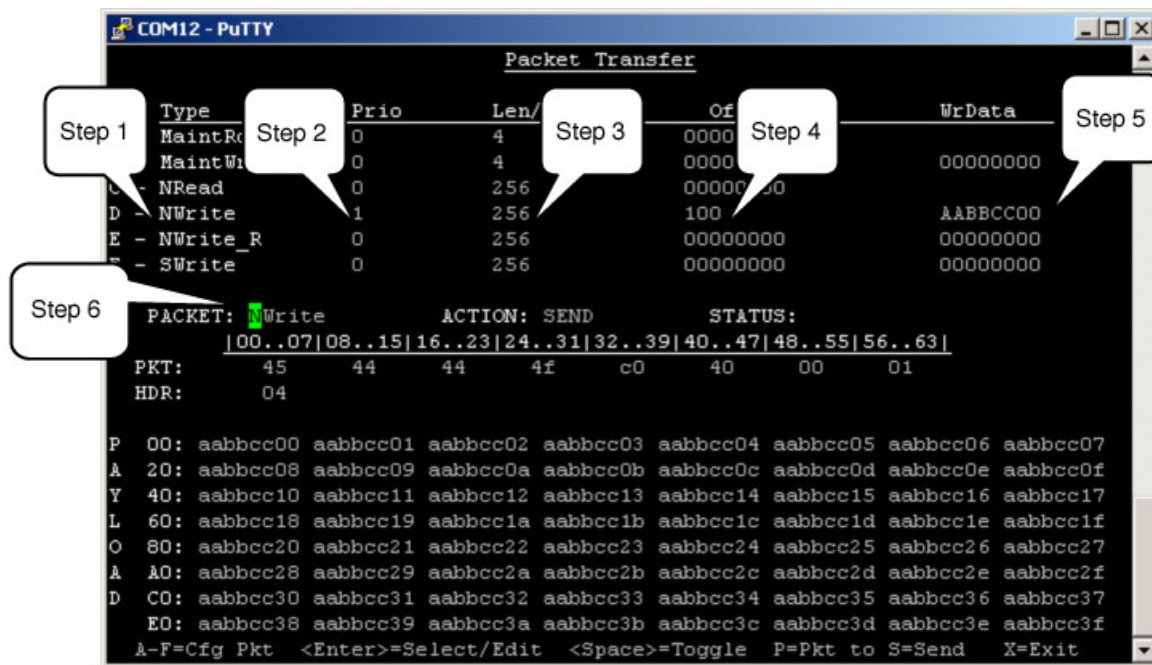
To begin the NWrite demo, first clear the contents of the Test Memory so that the NWrite transfer has a noticeable effect. From the Main Menu, select **V** to see the View Local Memory page and then press **C** to clear the contents. The View Local Memory page should look like Figure 22.

Figure 22. View Local Memory Page



Press **X** to return to the Main Menu and then press **P** to go to the Packet Transfer page. Figure 23 shows how to set up a 256-byte NWrite transfer to test memory offset 0x100 with an AABCC00 incrementing pattern.

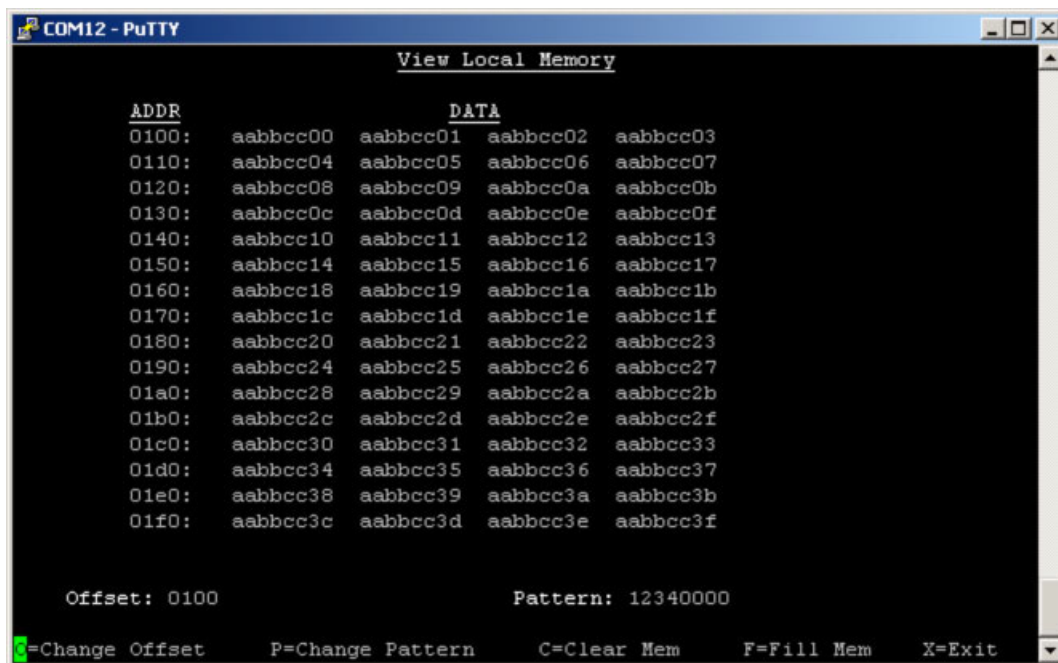
Figure 23. Setting up a 256-Byte NWrite Transfer



1. Press **D** to edit the NWrite packet configuration line.
2. Use the **Space** bar to toggle and select the priority at which the Maintenance Read should be sent. This example shows the selection of Priority 1. Press **Enter** to select.
3. The Length field is now editable. Press **Enter** to accept sending 256 bytes.
4. The Offset field is now editable. Enter **100** (hex) for the offset into the test memory to be written into.
5. The WrData field is now editable. Enter **AABBC000** for the pattern to be written into memory.
6. Press **P** to select the packet to send. Use the **Space** bar to toggle through the selections until NWrite is displayed and press **Enter** to select. The packet header is now displayed.
7. Press **S** to send the selected packet (NWrite).

After sending the NWrite packet, return to the View Local Memory page and confirm that only the memory in the address range 0x100 to 0x1FF has been modified with the selected pattern 0xAABBC00.

Figure 24 shows the results of the NWrite memory transfer.

Figure 24. NWrite Memory Transfer Results

Demonstrating Block Transfer Tests

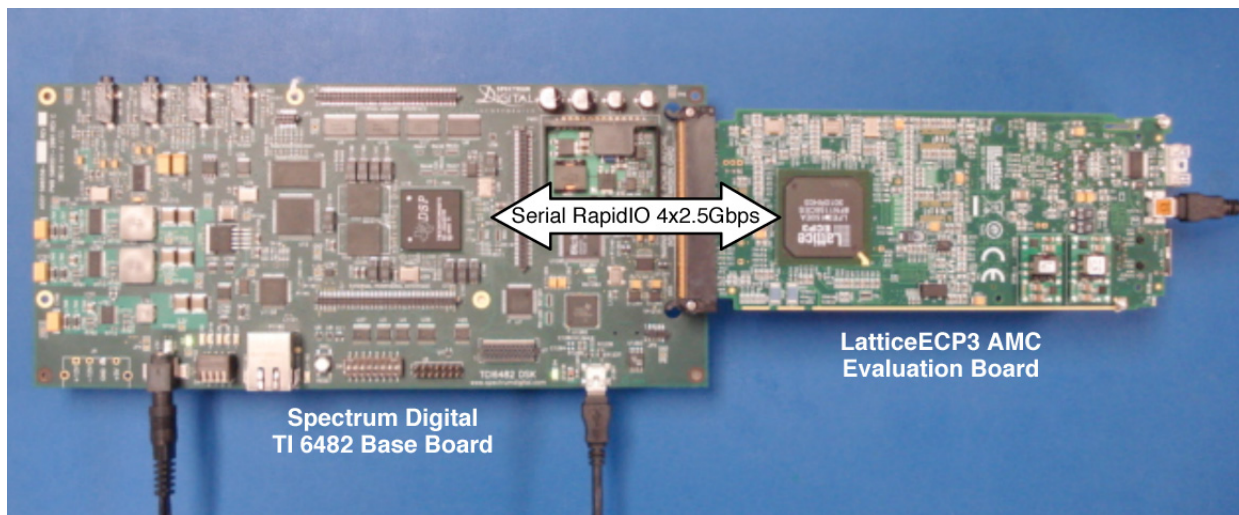
The InterOp Tests page allows the user to send large numbers of packets to/from the IP core to validate the robustness of the link and IP core. The InterOp Tests page is simple to operate. The only requirement is to enter the number of times the test should be repeated. Once the number of cycles is selected, press the number corresponding to the test number.

Things to observe:

- Use the SRIO IP Status page to examine the number of packets that have been sent and received. Look for error counts and error bits.
- Use the SRIO Register Display page to confirm the TAG CSR value is changing when running Test #2.
- Use the View Local Memory page to confirm that the contents of the SRIO test memory are changing when Tests #3 and #4 are run. These two tests perform NWrites into the test memory with a continuously incrementing 32-bit pattern so the data values will change from run to run.

Interoperation with Another Serial RapidIO Device

This section describes how the LatticeECP3 AMC Evaluation Board is configured to operate with another Serial RapidIO device using the loopback demo software. Figure 25 shows the LatticeECP3 AMC Evaluation Board installed in a Spectrum Digital base board. The Spectrum Digital base board contains a TI 6482 DSP processor with a 4x2.5Gbps Serial RapidIO link.

Figure 25. LatticeECP3 AMC Evaluation Board Installed in a Spectrum Digital Base Board

The LatticeECP3 AMC Evaluation Board is configured to access the TI DSP device over the Serial RapidIO link. The important configuration settings are the TI BaseID and the test memory address and size. These values need to match the external Serial RapidIO device's configuration.

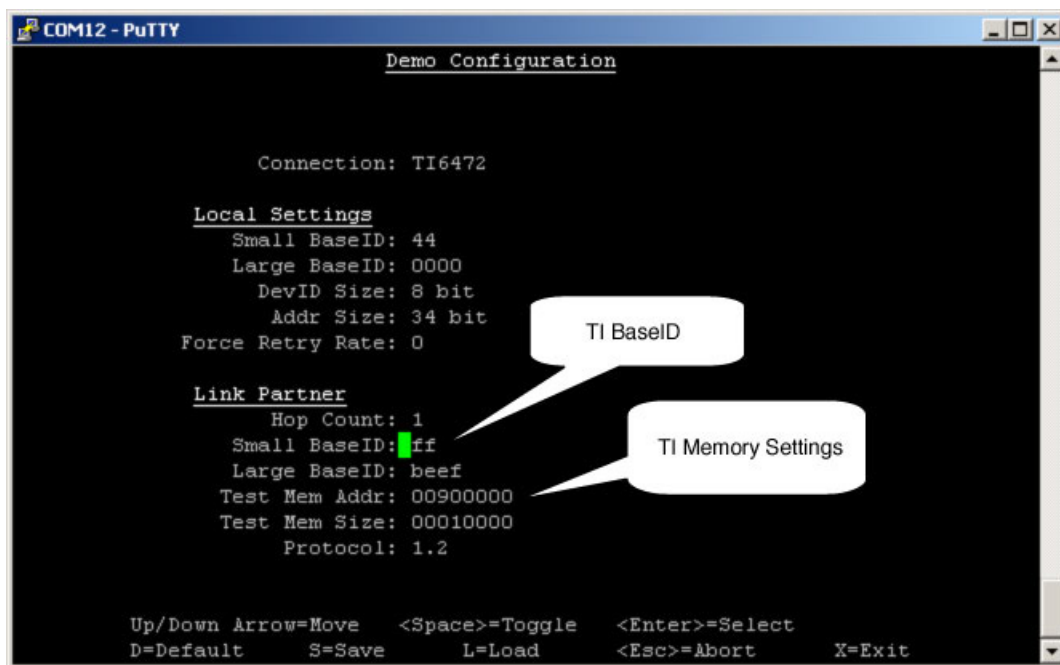
Figure 26. TI InterOp Configuration Settings

Figure 27 uses the remote SRIO Status Registers page to view the TI DSP Serial RapidIO configuration registers to verify maintenance accesses.

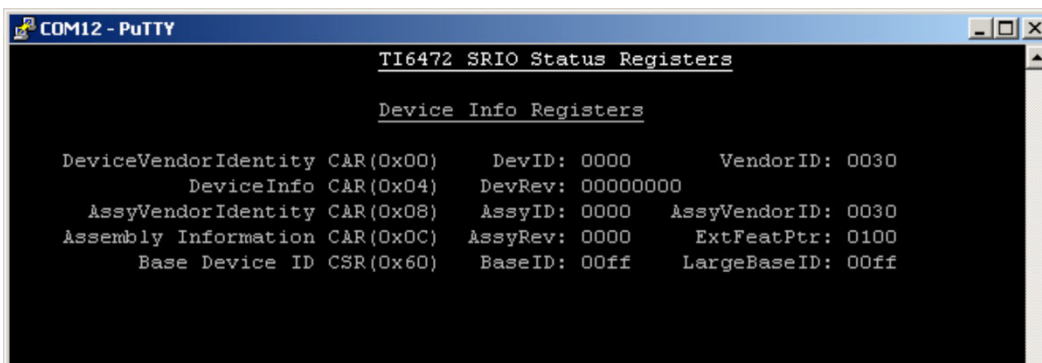
Figure 27. TI InterOp SRIO Status Registers Display

Figure 28 shows the running of a NWrite/NRead interoperability test to verify Serial RapidIO memory access to the TI DSP memory.

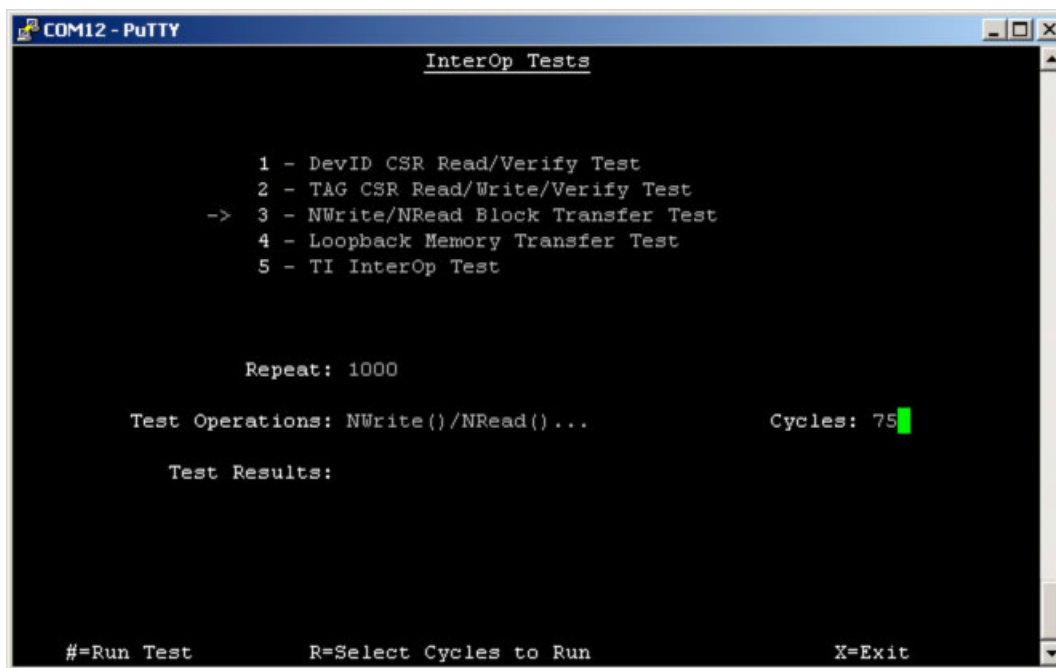
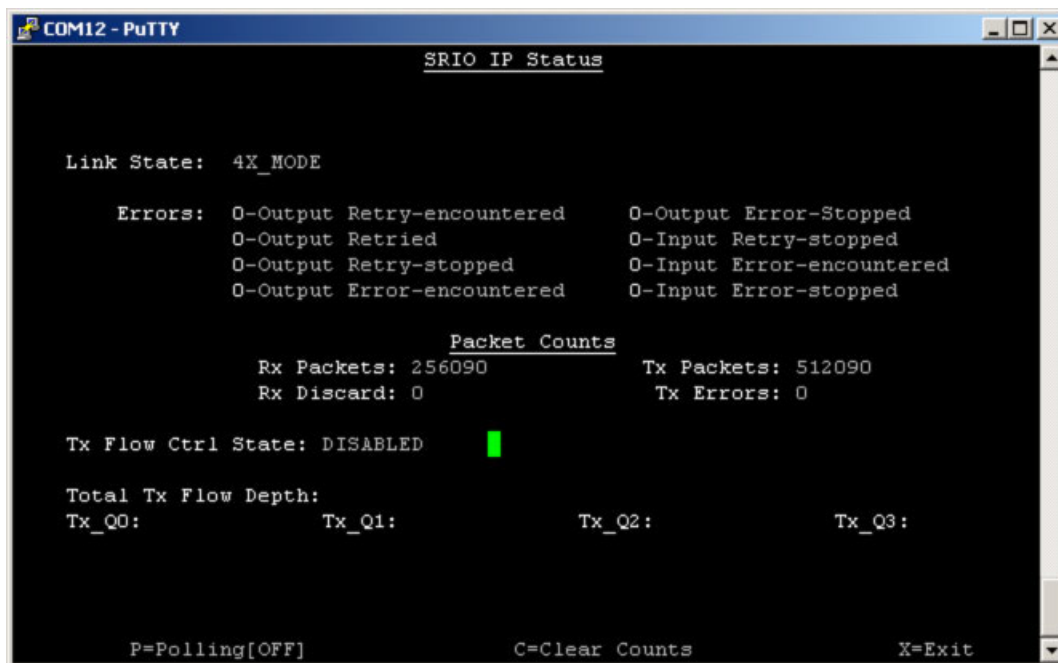
Figure 28. Running the TI InterOp Test

Figure 29 shows the SRIO IP Status page used to check for errors after the test is complete and to view the number of packets sent.

Figure 29. TI InterOp Test Results



Note that the number of transmitted packets is twice the number of received packets. This is correct for the NWrite/NRead test. Each NWrite transmits a packet, and each NRead transmits a packet and receives a packet. Therefore there are twice as many transmits as receives in this test.

Figure 30 shows the filling of 128 bytes of the TI 6482 DSP test memory with a data pattern 0x44332211 (note: the TI device is operating in Little Endian mode, so the values read back by the LatticeECP3 AMC Evaluation Board will be reversed, as the LatticeMico32 and Serial RapidIO are Big Endian protocol).

Figure 30. Modifying TI 6482 Memory Contents for NRead Access

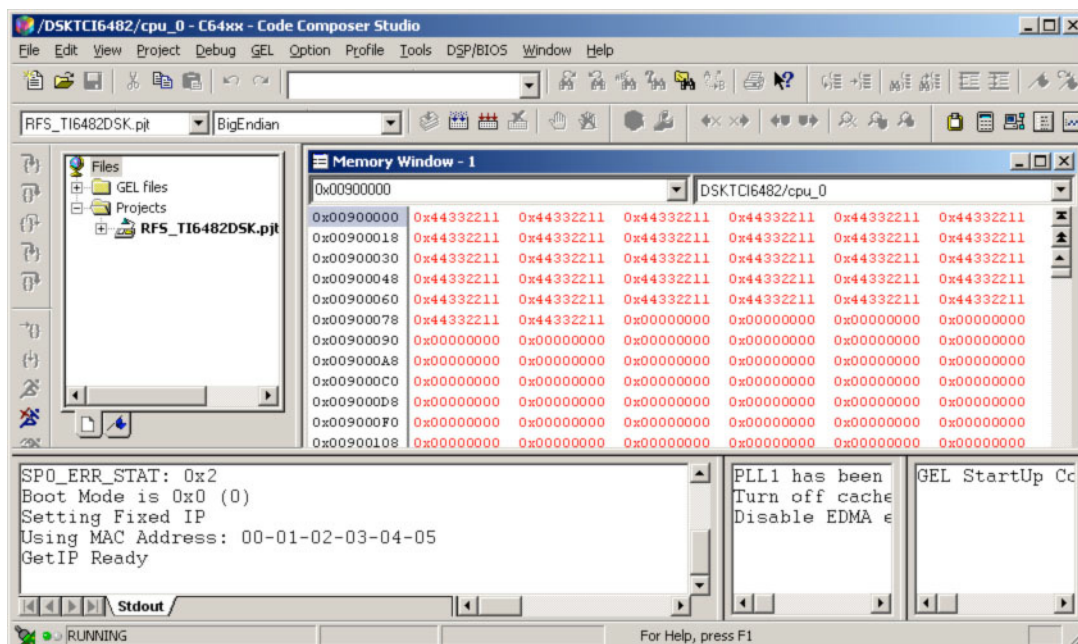


Figure 31 shows the results of reading back the TI memory using an NRead packet transfer to read back 256 bytes of the TI test memory starting at its base address (where the pattern was loaded). This confirms Serial RapidIO memory interoperability between the TI 6482 DSP and the RapidIO 2.1 Serial Endpoint IP core.

Figure 31. NRead Results from TI 6482

Type	Prio	Len/Size	Offset	WrData
A - MaintRd	0	4	00000000	
B - MaintWr	0	4	00000000	00000000
C - NRead	2	256	00000000	
D - NWrite	0	256	00000000	00000000
E - NWrite_R	0	256	00000000	00000000
F - SWrite	0	256	00000000	00000000

PACKET:	Read	ACTION:	RECV	STATUS:	OK
00..07 08..15 16..23 24..31 32..39 40..47 48..55 56..63					
PKT:	82	ff	44	4f	da 00 90 00
HDR:	04				

P 00:	11223344	11223344	11223344	11223344	11223344	11223344	11223344	11223344
A 20:	11223344	11223344	11223344	11223344	11223344	11223344	11223344	11223344
Y 40:	11223344	11223344	11223344	11223344	11223344	11223344	11223344	11223344
L 60:	11223344	11223344	11223344	11223344	11223344	11223344	11223344	11223344
O 80:	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
A A0:	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
D C0:	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
E0:	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

A-F=Cfg Pkt <Enter>=Select/Edit <Space>=Toggle P=Pkt to S=Send X=Exit

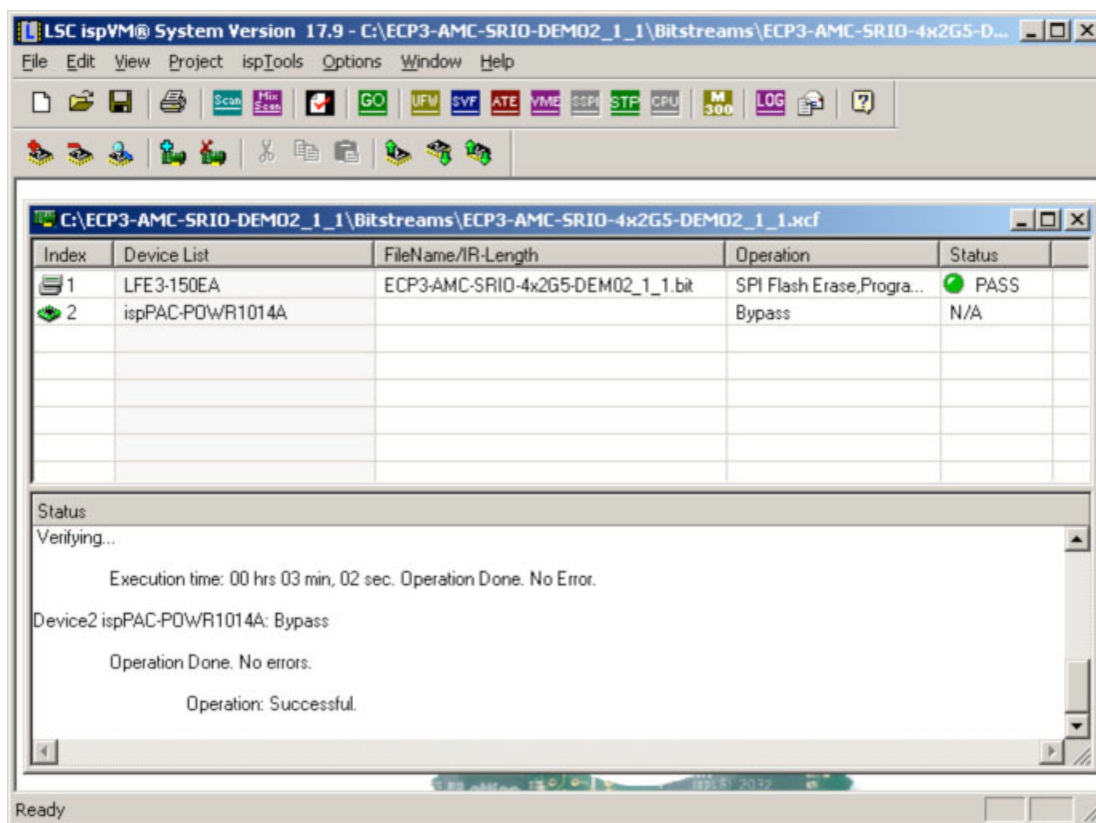
Upgrading the Demo Bitstream/Software

To upgrade the demo bitstream, obtain the new bitstream from the [LatticeECP3 Serial RapidIO Solutions web page](#) on the Lattice web site. Connect the ispVM download cable to the JTAG header on the AMC Interface Card as shown in EB56, [LatticeECP3 AMC Evaluation Board - Revision B User's Guide](#).

Important: Do not follow the instructions for using the BSCAN2 procedure for loading a bitstream into the SPI Flash. This mode of operation is not used for this demo configuration. The LatticeECP3 is directly in the JTAG scan chain.

Open and use the provided ispVM XCF configuration file to set up the ispVM System software and load the proper bitstream file.

Program the SPI Flash by pressing the **GO** button. Figure 32 shows an example of a bitstream that has been successfully programmed into the SPI Flash on a LatticeECP3 AMC Evaluation Board. Don't forget to press **PROGRAM** or power cycle the board to load the new bitstream image into the LatticeECP3.

Figure 32. LatticeECP3 AMC Bitstream Programming with ispVM

References

- IPUG84, [RapidIO 2.1 Serial Endpoint IP Core User's Guide](#)
- EB56, [LatticeECP3 AMC Evaluation Board - Revision B User's Guide](#)

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

Revision History

Date	Version	Change Summary
November 2010	01.0	Initial release.