PCI Express-to-DDR3 SDRAM Reference Design

© July 2009 AN-431-1.1

Introduction

The Altera® PCI Express to DDR3 SDRAM reference design provides a sample interface between the Altera PCI Express MegaCore® function and a 64-bit, 512 MByte DDR3 SDRAM memory. Altera offers this reference design to demonstrate the operation of the PCI Express MegaCore function and the DDR3 SDRAM High-Performance Controller MegaCore function. The reference design has the following features:

- Supports PCI Express (PCIe) endpoint to PCIe root complex DMA read and write transactions
- Uses the PCI Express hard IP MegaCore function
- Uses the DDR3 SDRAM High Performance Controller MegaCore function
- Uses a Stratix® IV GX device with internal transceivers

This document contains the following topics:

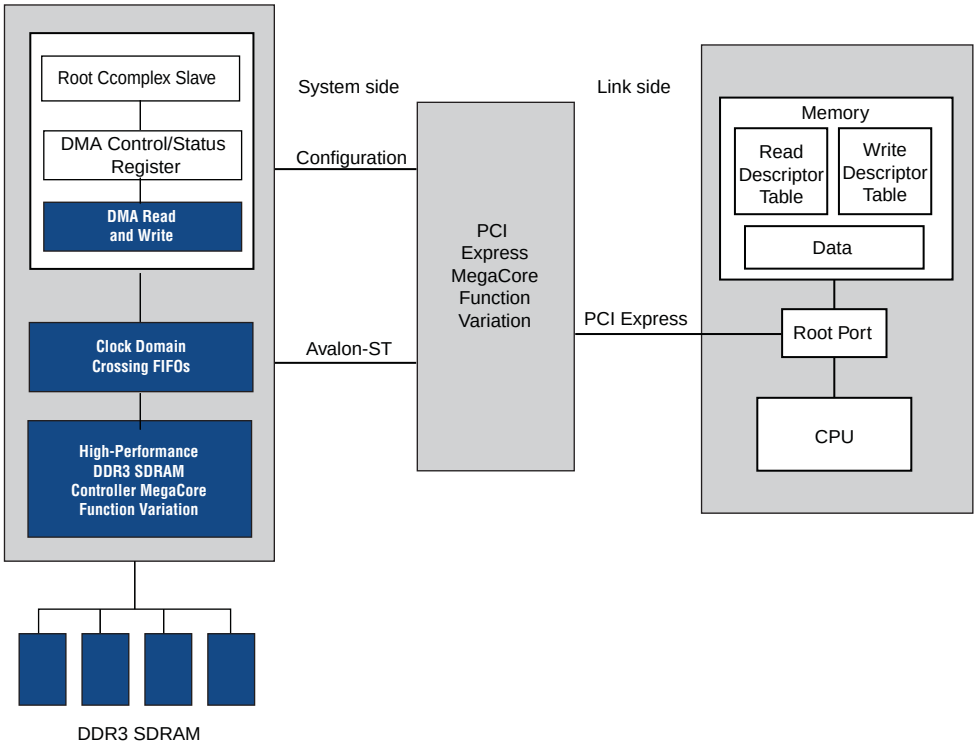
Reference Design Overview

The reference design connects the Altera PCI Express MegaCore function to DDR3 SDRAM memory using the reference design interface circuitry. The design runs on Altera's Stratix IV GX FPGA Development Kit, which includes a PCI Express development board. Altera also provides a software driver, programming information, and a GUI to run the application.

The Altera PCI Express to DDR3 SDRAM reference design interfaces to the system side of the Altera PCI Express MegaCore function. (Refer to Figure 1 on page 2.) The external side of the PCI Express MegaCore function forms half of the PCI Express link. The high-performance DDR3 SDRAM controller accesses external DDR3 SDRAM memory. The PCI Express MegaCore function generally operates as a PCIe master, or initiator. When the PCI Express MegaCore function operates as a PCIe master, the DMA engine initiates the transaction, monitors the status, and manages the progress of the data transfers.

The Altera PCI Express to DDR3 SDRAM reference design is very similar to the chaining DMA design example that you automatically generate when you create a PCI Express Megacore function using the MegaWizard Plug-In Manager. Refer to "*Chapter 7 Testbench and Design Example*" in the *PCI Express Compiler MegaCore Function User Guide* for a detailed description of the chaining DMA design example.

Figure 1. Block Diagram for the PCI Express to DDR3 Reference Design



The difference between the chaining DMA design example and the PCI Express to DDR3 SDRAM reference design is that the reference design uses an external DDR3 memory to store data instead of internal memory in the FPGA. Consequently, the files in the chaining DMA design example that define the DMA controller and memory accesses are modified. In addition, the top-level Verilog HDL file, ***<name>*_example_chaining_top.v**, points to the DDR3 version of these files.Table 1 lists the files in the DMA design example and PCI Express to DDR3 SDRAM reference design that access memory.

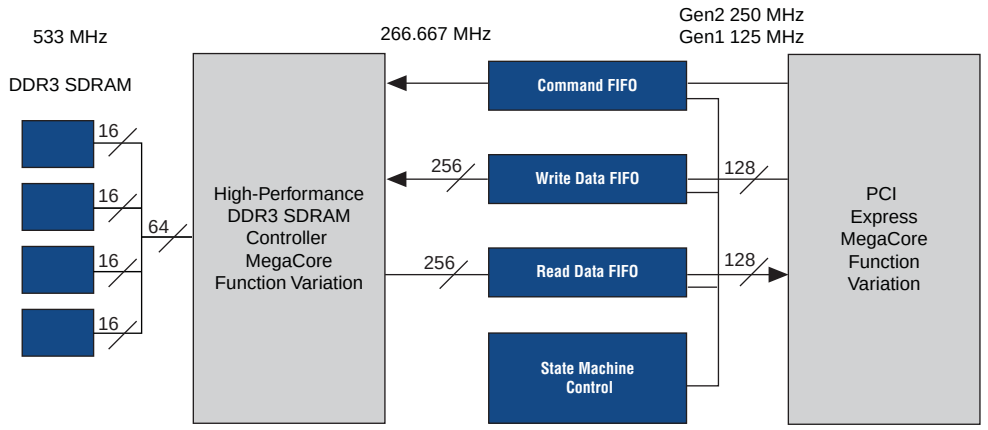
Table 1. Files that Access Memory

File Name in Chaining DMA Design Example	File Name in PCI Express to DDR3 SDRAM Reference Design
altpcierd_write_dma_requester_128.v	altpcierd_write_dma_requester_128_ddr.v
altpcierd_dma_dt.v	altpcierd_dma_dt_ddr.v
altpcierd_dma_prg_reg.v	altpcierd_dma_prg_reg_ddr.v
altpcierd_cdma_app_icm.v	altpcierd_cdma_app_icm_ddr.v
altpcierd_example_app_chaining.v	altpcierd_example_app_chaining_ddr.v
*<name>*_example_chaining_pipen1b.v	*<name>*_example_chaining_pipen1b_ddr.v

Figure 1 highlights these modules that include these files in dark blue.

The following sections provide an overview of changed modules. Figure 2 shows the datapath that these modules implement.

Figure 2. Datapath from PCI Express MegaCore Function to DDR3 SDRAM



PCIe to DDR3 Interface Block

The DDR3 memory transfers 64 bits on each clock edge with a burst size of 8 for a total of 512 bits per burst transfer. The local interface of the DDR3 high performance controller is 256 bits wide and runs at half rate, or 266.667 MHz, so that the local interface must complete two transfers to match the DDR3 transfer size of 512 bits.

The reference design uses three FIFOs to pass data between the DDR3 interface clock domain and the PCI Express clock domain. The three separate FIFOs handle access commands, write data, and read data. The FIFOs are instantiated in a block called **DDR3_Fifo_Interfaces.vhd**. This file also defines a state machine to control the timing of the data and commands to the DDR3 controller's local interface and monitor the status signals from it.

The following pseudo code summarizes the algorithm that the state machine implements:

1. Wait for the command FIFO to not be empty.
2. Decode the read or write command.
3. For reads, ensure the DDR3 high performance controller is ready to receive a command, then send the read.
4. For writes, because the PCI Express word size is 128 bits, retrieve 4 consecutive words before sending two, 256-bit words on consecutive clock cycles. To match throughput, four write commands are required by the PCIe MegaCore function for each DDR3 write in this particular design application. The write commands must be to consecutive addresses. These constraints are built into the state machine. When all data is collected and the DDR3 controller is ready, the state machine writes the data to the DDR3 controller and then decodes the next command.

There are several alternatives when designing hardware that interfaces to the DDR3 SDRAM High Performance Memory Controller. For more information refer to the *DDR3 SDRAM High Performance Controller User Guide*.

DDR3 SDRAM IP Block

The Stratix IV GX FPGA Development Kit uses a bank of four, 16-bit Micron MT41J64M16 DDR3 components to create a 64-bit memory interface. The interface uses fly-by topology as specified in *JEDEC Standard for DDR3 SDRAM, JES79-3C*. The entire interface runs at the maximum supported speed of 533 MHz to ensure that the DDR3 memory system does not limit the bandwidth of the reference design. Example 1 gives the bandwidth calculations for the PCI Express and DDR3 SDRAM interfaces.

Example 1. Bandwidth Calculations

```
Gen2 PCI Express - 128 bits @250 MHz = 32 Gbps (minus some overhead)
Gen1 PCI Express - 128 bits @125 MHz = 16 Gbps (minus some overhead)
DDR3 - 64 bits @ 533 MHz = 68.4 Gbps (minus overhead due to controller and interface
       inefficiences)
```

As these calculations indicate, by running the DDR3 interface at 533 MHz, the available bandwidth easily accommodates the bandwidth required for Gen2 operation. The reference design uses Altera's DDR3 high performance controller which consists of a memory PHY (ALTMEMPHY) and controller code.

For more information on using DDR3 SDRAM with Stratix III and Stratix IV family FPGAs, refer to *AN 436: Using DDR3 SDRAM in Stratix III and Stratix IV Devices*. For more information on the high performance DDR3 SDRAM controller, refer to the *DDR3 SDRAM High Performance Controller User Guide.*

Using the Reference Design

This section describes how to install the reference design and provides instructions for running the software application. The following information is included:

- Hardware Requirements
- Software Requirements
- Software Installation
- Hardware Installation
- Running the Software Application

Hardware Requirements

The reference design requires the following hardware:

- The Stratix IV GX FPGA Development Kit.
- A computer running 32-bit Windows XP with an ×8/x4/×1 PCI Express slot for the Stratix IV GX FPGA development board. The software application and hardware are installed on this computer, referred to as computer #1 in this document.
- A computer with the Quartus II software for downloading FPGA programming files to the Stratix IV GX FPGA development board, referred to as computer #2 in this document.

- A USB cable or other Altera download cable.

Software Requirements

To run the reference design application requires installation of the following software:

- Reference design software installed on computer #1.
- PCI Express to DDR3 reference design package, available as a downloadable compressed file.

  For more information refer to the PCI Express to DDR3 SDRAM Reference Design product page.

- The Quartus II software version 9.0 running on computer #2.

Software Installation

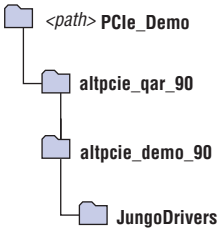
You must have Administrator privileges to install the software application.

The software application only runs on 32-bit Windows XP and includes the WinDriver software from Jungo as part of the application.

Perform the following steps to install the software application and Windows drivers:

1. Download the PCI Express to DDR3 reference design package to computer #1, and extract the compressed files. Figure 3 shows the directory structure.

Figure 3. Directory Structure



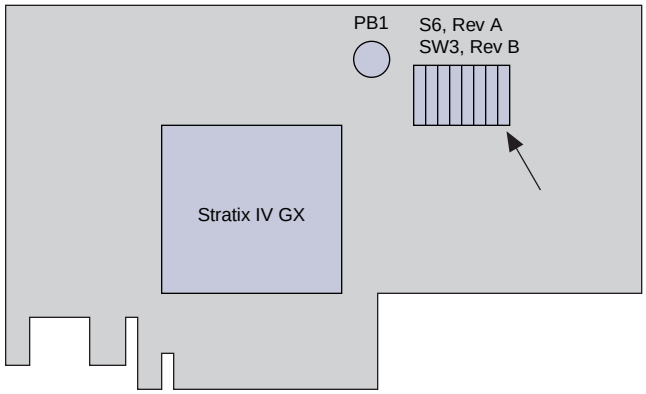
2. Before plugging in the PCI Express card, copy the **altpcie_demo_90** directory to computer #1.
3. In the **JungoDrivers** directory, double-click on **install.bat** to install the Windows XP driver for this application.
4. Run **altpcie_demo.exe** from the **altpcie_demo_90** directory to start the software application.

Hardware Installation

Before you plug the Stratix IV GX card into computer #1, you must check the settings on an eight-position dip switch which controls the PCI Express mode of operation. Figure 4 highlights this component. The right-most position of this dip switch is used to change between normal operation and PCI Express compliance base board (CBB) testing. To run the software included in this application note, this switch must be in the **off** position. When it is in the **on** position, you can use the reset switch labeled PB1 to cycle through various modes required for CBB testing. (The dip switch labels the **on** side on the switch.)

The top-level RTL file has been modified to enable CBB testing. If you regenerate the MegaCore function, you may overwrite this top-level file and disable the CBB testing capability.

Figure 4. Location of Components that Control PCI Express Mode of Operation



Perform the following steps to install the hardware.

1. Power down computer #1 and plug the Stratix IV GX card into the PCI Express slot. Depending on your hardware, a PCI Express lane converter may be required.
2. The Stratix IV GX FPGA Development Kit includes integrated USB-Blaster circuitry for FPGA programming. However, for the host computer and development board to communicate, you must install the USB-Blaster on the host computer.

   To download the USB-Blaster driver, go to the Altera support site at **www.altera.com/support/software/drivers/dri-index.html**. For installation instructions, go to **www.altera.com/support/software/drivers/usb-blaster/dri-usb-blaster-xp.html.**
3. Program the FPGA with the reference design using the Quartus II software on computer #2 and an Altera USB-Blaster cable (or other download cable) connection between computer #2 and the development board on computer #1.

Connecting the USB-Blaster Cable

To connect the USB-Blaster cable, perform the following steps:

1. Connect one end of the USB cable the USB port (J7) on the Stratix IV GX FPGA development board.

2. Connect the other end of the cable to the USB port on the computer running the Quartus II software on computer #2.

Programming with the .sof File

To program the board with the **.sof** file provided, interrupt the boot sequence on computer #1 to bring up the BIOS System Setup interface. (Pressing the F2 key interrupts the boot sequence on many Windows PCs.)

Perform the following steps to program the FPGA with the **.sof** file:

1. Start the Quartus II programmer on computer #2.
2. Click **Hardware Setup** and select the **USB Blaster**. Click **Close**.
3. In the Quartus II Programmer, click **Auto Detect** to list the devices attached to the JTAG chain on the Stratix IV GX FPGA development board.
4. Right-click the Stratix IV GX device (EP4SGX230) and click **Change File**. Select the path to the appropriate **.sof** file.
5. Turn on the **Program/Configure** option for the added file.
6. Click **Start** to download the selected file to the Stratix IV GX device. The device is configured when the **Progress** bar reaches 100%.
7. On computer #1 exit the BIOS System Setup or boot manager interface.
8. On computer #1, press `Ctrl+Alt+Delete` to perform a soft reboot.
9. The operating system detects a new hardware device and displays the Found New Hardware Wizard. In the wizard, select **Install the software automatically (Recommended)**. Click **Next**.
10. Click **Finish** to close the wizard.

Running the Software Application

Perform the following steps to run the software application:

1. Double-click on the application **altpcie_demo.exe** in the **altpcie_demo_90** directory.
2. The application reports the board type, the number of active lanes, the maximum read request size, and the maximum payload size.

   The software GUI has the following control fields:

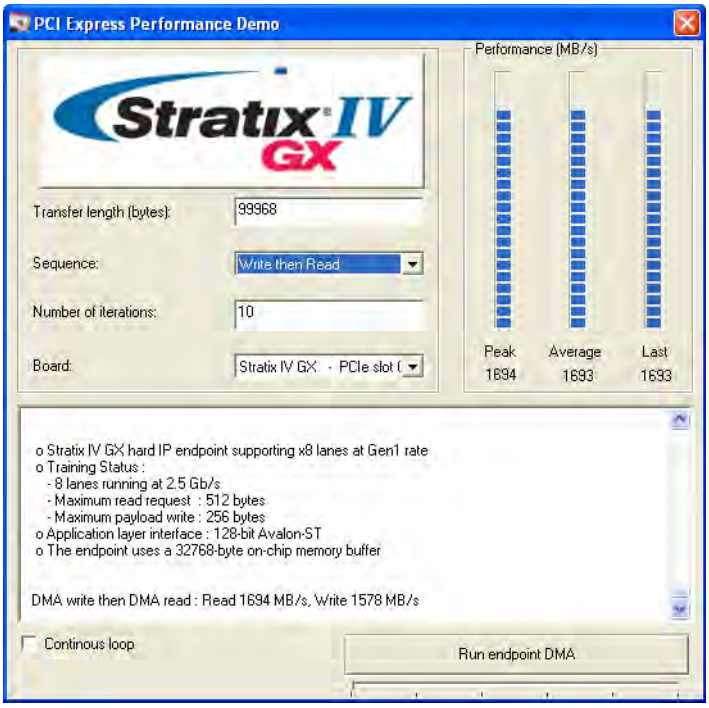
   - **Transfer length**—Specifies the transfer length in bytes
   - **Sequence**—Controls the sequence for data transfer or addressing
   - **Number of iterations**—Controls the number of iterations for the data transfer
   - **Board**—Specifies the development board for the software application
   - **Continuous loop**—When this option is turned on, the application performs the transfer continuously

3. Set the **Transfer length** to 100,000 bytes and the **Sequence** to **Write then Read**, Click **Run**.

   When set for **Write then Read**, the software programs the DMA registers in the FPGA to transfer data from the FPGA to the DDR3 SDRAM in chunks of 100,000 bytes and then read it back. The performance bars report the peak, average, and last throughput. The average throughput is computed across all the iterations.

4. You can use the GUI to change the **Transfer length** and **Sequence** and repeat the test.

Figure 5. Write then Read Options



In addition to the parameter settings to control the chaining DMA, the GUI includes additional commands that you can use to obtain configuration information about the device and board and to perform root port reads and writes. Table 2 outlines all of the available commands. The position of the slider control changes the command.

Table 2. PCI Express Performance Demo GUI Commands and Options (Part 1 of 2) *(Note 1)*

Command	Options	Description
Run endpoint DMA (Figure 5)	**Write only** **Read only** **Read then write** **Write then read**	Writes transfer data from the FPGA to system memory. Reads transfer data from system memory to the FPGA.
Scan the endpoint configuration space registers (Figure 6)	**Type 0 Configuration** **PCI Express capability** **MSI capability** **Power management capability**	Reports the byte address offset, value and a description of the selected register set.
Scan the current PCI Express board settings (Figure 7)	—	Reports the configuration settings of the Stratix IV GX FPGA development board.

Table 2. PCI Express Performance Demo GUI Commands and Options (Part 2 of 2) *(Note 1)*

Command	Options	Description
Scan the motherboard PCI bus	—	Reports the vendor ID, device ID, slot, bus, and function numbers for all devices on the motherboard's PCI bus.

Notes to Table 2:

(1) This software application is a different version of the software application used in *Application Note 456: PCI Expre ss High-Performance Reference Design*. To run this reference design, you must use the software version included with this reference design.

Figure 6 illustrates the output of the **Scan the current PCI Express board setting s** command.

s

Figure 6. Scan the Current PCI Express Board Settings

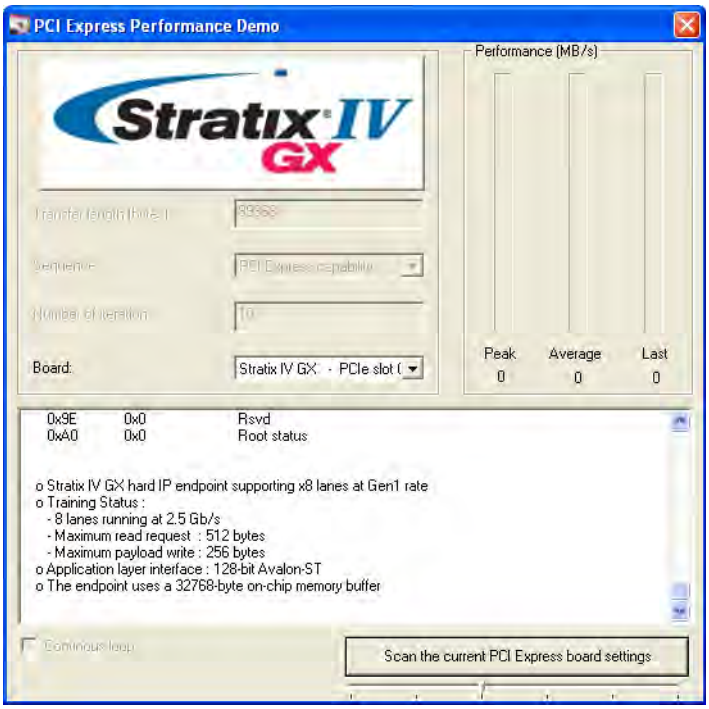


Figure 7 illustrates the output of the **Scan the motherboard PCI bus** command.

Figure 7. Scan the Motherboard PCI Express Bus

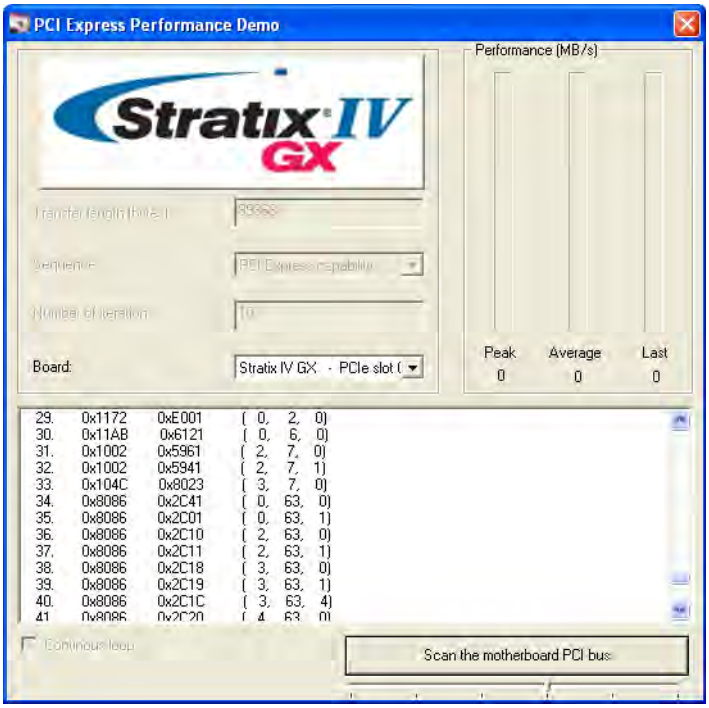


Table 3 gives the performance of the Stratix IV GX FPGA development board using this reference design with a motherboard that includes the Intel X58 Chipset. The table shows the average throughput for a transfer size of 100 KBytes and 20 iterations with a maximum write payload size of 256 bytes, a maximum read request size of 512 bytes, a read completion size of 256 bytes using a clock multiplier unit (CMU) clock.

Table 3. Stratix IV GX Performance - Intel X58 Chipset

Configuration	DMA Reads (MB/s)	DMA Writes (MB/s)
Gen1 ×8 128b interface	1613	1695

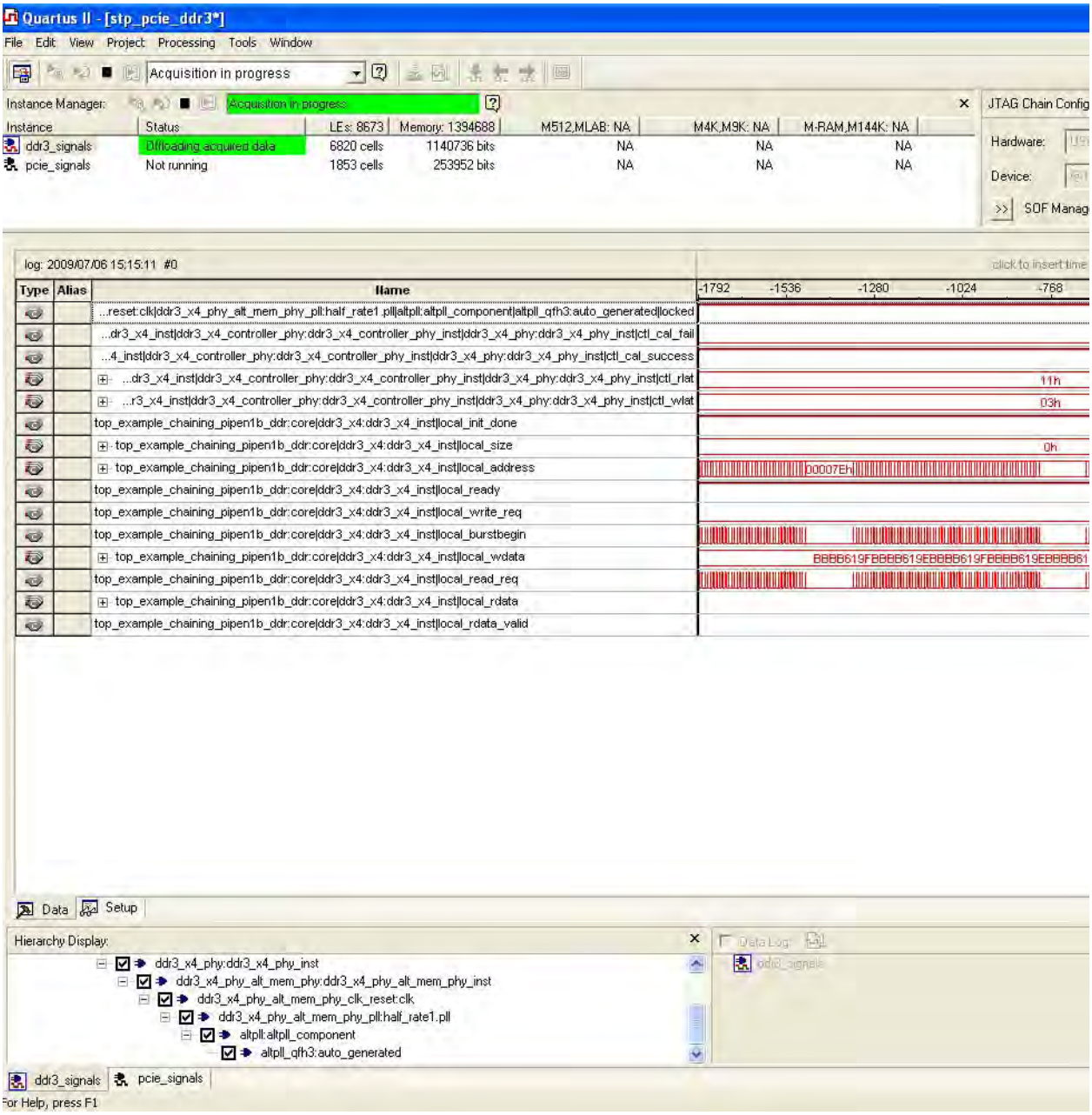
SignalTap II Files

The reference design package also includes SignalTap II files (**.stp)** that you can use with the SignalTap II Embedded Logic Analyzer to obtain information on the performance of this design. The SignalTap II files includes the key signals from the application logic. The `init` signal in the DMA read and write modules transitions to zero at the beginning of the transfer. You can use the `init` signal as a trigger in the SignalTap II file to capture data.

The `tx_st_ready0` and `rx_st_valid0` are indications of link utilization and throughput. In the transmit direction, the frequent deassertion of the `tx_st_ready0` signal typically indicates that the MegaCore function is not receiving enough credits from the device at the far end of the PCI Express link. It could also indicate that a wider link has trained to a narrower size. In the receive direction, the deassertion of `rx_st_valid0` indicates that the MegaCore function is not receiving enough data.

Figure 8 shows an example of the SignalTap II Embedded Logic Analyzer GUI.

Figure 8. The SignalTap GUI



Design Limitations

This reference design has the following limitations:

- The DMA engine is not designed to transfer data with a size of less than 64 bytes.
- The DMA engine only transfers data in multiples of 64 bytes and the data must be aligned on a 64-byte address.
- The root port must use a DMA to access the SDRAM memory.

References

This application note references the following documents:

- *AN 436: Using DDR3 SDRAM in Stratix III and Stratix IV Devices*
- *AN 456: PCI Express High-Performance Reference Design*
- *DDR3 SDRAM High Performance Controller User Guide*
- *JEDEC Standard for DDR3 SDRAM, JES79-3C*
- *PCI Express Compiler MegaCore Function User Guide*

Revision History

Table 4 shows the revision history for this application note.

Table 4. Template Revision History

Date and Revision	Changes Made	Summary of Changes
July 2009, version 1.1	Updated to use the PCI Express hard IP implementation with DDR3 SDRAM.	Updated to showcase newer technology.
August 2006, version 1.0	Initial Release	—

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support