|  |  |  |  |
| --- | --- | --- | --- |
| [ |  | ] |  |
| **InfiniBand FPGA Test Harness Architecture** |
| Senior Design May-0904 Project Team  Matt Wall  Tim Prince  Ryan Schwarzkopf  Xiang Li  Alex Burds | | | |
| Abstract | | | |
| The purpose of this document is to give a brief overview of the design of the InfiniBand test harness being developed by the May0904 project team for the EE/CprE 492 senior design course at Iowa State University. For more information about the InfiniBand FPGA project, see  http:// seniord.ece.iastate.edu/may0904 | | | |
| Disclaimer | | | |
| This document was developed as part of the requirements of an electrical and computer engineering course at Iowa State University, Ames, Iowa. The document does not constitute a professional engineering design or a professional land surveying document. Although the information is intended to be accurate, the associated students, faculty, and Iowa State University make no claims, promises, or guarantees about the accuracy, completeness, quality, or adequacy of the information. Document users shall ensure that any such use does not violate any laws with regard to professional licensing and certification requirements. Such use includes any work resulting from this student prepared document that is required to be under the responsible charge of a licensed engineer or surveyor. This document is copyrighted by the students who produced the document and the associated faculty advisors. No part may be reproduced without the written permission of the senior design course coordinator. | | | |

# Introduction

The InfiniBand specification is a two-part document consisting of over a thousand pages. As one might expect, verifying that a particular implementation meets these specifications can be an difficult, time consuming and expensive process.

As part of our project here at Iowa State University, we have developed open source VHDL implementations of the InfiniBand data link and physical layers. With initial implementation and preliminary verification nearing completion, we are moving on to addressing the issue of full scale verification.

To that end, we are now beginning to develop a full-scale test harness which will allow a future development team to verify and expand upon our implementation in a straightforward manner.

# System Architecture

Figure 1provides a top level diagram of the test harness architecture. A PowerPC 440 core on a Virtex-5 FPGA is interfaced to one or more InfiniBand test harnesses over the IBM Processor Local Bus (PLB). Each test harness contains an instance of the InfiniBand link and physical layers, as well as an instance of the Xilinx RocketIO gigabit transceiver IP core.



Figure - System Diagram

# Test Harness Architecture

The test harness itself will include an instance of the InfiniBand link and physical layers, a PLB control interface, and a variety of programmable interconnects to provide the maximum possible level of flexibility for the user.

Figure 2 depicts the basic architecture of the test harness. The PLB interface connects the control interface of the test harness to the PLB, the control register allows the user to program the various interconnects and control the operation of the test harness, and the TX and RX FIFOs allow the user to inject and extract data into and out of the datapath at various points.



## PLB Interface

The PLB interface provides handles the translation of PLB signals into a format compatible with the test harness interface. The test harness appears to the PLB as 3 memory mapped registers. The memory map for the test harness is given in appendix A.

## Control Register

The control register is the primary control interface for the test harness. The layout of the register and a description of the different fields are given in appendix B.

## RX and TX FIFOs

The RX and TX FIFOs allow the user to inject and extract data to and from various points on the datapath.

To inject data into a point on the datapath, the user first fills the FIFO buffer with the data to be injected, then sets the TX tap point in the control register. The data from the FIFO will be injected into the datapath at a rate of one word per cycle until the FIFO is empty. Once the FIFO is empty, the TX tap point is automatically reset so that no data is injected into the datapath. While data is being read from the FIFO, the user may continue to add data to the FIFO.

To extract data from the datapath, the user configures the RX tap point to the appropriate position. Data from the datapath will then be written to the RX FIFO at a rate of one word per cycle until the RX FIFO is full, at which point the oldest data in the FIFO is ejected. The end result is that the RX FIFO contains the last 8k values of the connected bus.

Figure - Test Harness Diagram

# Appendix A – Memory Map

|  |  |  |
| --- | --- | --- |
| Index from base address | Mapped Object | Description |
| +0x00 | SCR | Status and control register |
| +0x04 | TX\_WR\* | TXFIFO write register |
| +0x08 | RX\_RD\*\* | RXFIFO read register |

*\* Register is write-only*

*\*\* Register is read-only.*

# Appendix B – Status and Control Register

TBD.



Figure - Control Register Layout