PCI Interfaces on DØ Silicon Trigger Card

E. Hazen, W. Earle - Boston University
February 8, 2001

Abstract
The DØ STT silicon trigger card (STC) logic mezzanine board contains three independent PCI interfaces. Each is likely to be implemented in a separate Altera FPGA. This document describes the functionality of each in detail.

This is a preliminary document and should not be used for reference!

1 PCI Interface 1
This is a master/target 32-bit 33MHz interface to PCI bus 1 on the motherboard. It’s main function is to retrieve the road broadcast information from the FRC via a Link Receiver Board (LRB). Data is presented to the STC logic as 32 bit words plus “data valid” and beginning and end of event marks.

A block diagram of the interface is shown in Figure 1.

![Figure 1: PCI Interface 1](image)

The control logic monitors the state of the DAV input. When this signal goes active, it initiates a PCI block transfer from the BAR0 (data) address of the LRB. Normally this transfer will read a complete event block from the LRB and write it via the “User Side” signals to the STC logic.

The target portion of the interface has one register (accessible in configuration space) which is used to hold the BAR address of the LRB which is to be accessed. This register is written once by the CPU during PCI configuration.

more details of interacting with the LRB needed...
1.1 “User Side” Signals

**DATA[31..0]**
PCI data from LRB. The header and trailer words from the LRB are included. The detailed data format is beyond the scope of this document (see FRC documentation). The Data changes on the falling edge of CLOCK and is stable at the rising edge.

**HEADER, TRAILER**
indicate that the current data word is either the first or last word of an event, respectively. These signals have the same timing as the data.

**WRITE**
indicates that the DATA, HEADER and TRAILER signals are valid within the current clock period. This signal has the same timing as the data.

**CLOCK**
Periodic clock derived from the PCI bus (nominally about 33MHz). All other signals are stable well before and after the rising edge of CLOCK.

1.2 “PCI Side” Signals

**PCI Bus**
implements all standard PCI signals for a 32-bit bus. More details of PCI functionality are given below.

**DAV**
is a “Data Valid” signal from the LRB, carried on a user-defined signal pin. This signal indicates that there is data available for reading in the LRB FIFO.

2 PCI Interface 2

This is a master/target interface to PCI bus 2 on the motherboard. It’s main function is to write data to a Link Transmitter Board (LTB) which transmits data to the Track Fit Card (TFC).

A block diagram of the interface is shown in Figure 2.

![Block Diagram of PCI Interface 2](image)

Figure 2: PCI Interface 2

The data is written to a FIFO, which is assumed to be large enough that a “full” condition is a fatal error (SCL.INIT in DØ). The control logic reads words from the FIFO as they become available.
available and determines whether each word is a HEADER, DATA or TRAILER word by examining bits 29–31, with specific bit assignments T.B.D.. HEADER and TRAILER words are transmitted using single-word PCI transfers. DATA words are transmitted using a PCI block transfer, which is terminated when either the FIFO becomes empty or a TRAILER word is seen.

The target portion of the interface has one register (accessible in configuration space) which is used to hold the BAR address of the LRB which is to be accessed. This register is written once by the CPU during PCI configuration.

The FIFO may be either an external FIFO or an internal FIFO implemented using RAM on the FPGA.

2.1 “User Side” Signals

<table>
<thead>
<tr>
<th>SIGNAL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>32 bit input data</td>
</tr>
<tr>
<td>FULL flag</td>
<td>FIFO full flag. FIFO is by design large enough that if this flag comes on it is an SCL_INIT condition.</td>
</tr>
<tr>
<td>WRITE</td>
<td>Indicates that valid data is present at the DATA pins</td>
</tr>
<tr>
<td>CLOCK</td>
<td>Periodic clock provided by user logic. All other signals must be stable before and after the rising edge of CLOCK.</td>
</tr>
</tbody>
</table>

2.2 “PCI Side” Signals

The PCI side implements standard PCI signals only.

3 PCI Interface 3

This is a target only interface to PCI bus 3 on the motherboard. It provides two main functions: a download/monitoring interface to all other FPGAs on the STC logic board, and an interface to the Level 3 data FIFO.

A block diagram of the interface is shown in Figure 3.

3.1 Level 3 FIFO Interface

<table>
<thead>
<tr>
<th>SIGNAL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA[31..0]</td>
<td>Level 3 Data. HEADER, DATA, TRAILER word types are identified by specific bit patterns.</td>
</tr>
<tr>
<td>EMPTY</td>
<td>FIFO Empty flag</td>
</tr>
<tr>
<td>READ</td>
<td>FIFO Read signal</td>
</tr>
<tr>
<td>CLOCK</td>
<td>Clock output to FIFO.</td>
</tr>
</tbody>
</table>

The FIFO interface is used to read data for Level 3. It is a full-speed interface (32MHz transfer rate) with a dedicated data input path of 32 bits. HEADER, DATA and TRAILER words are sensed by examining bits 29–31, with specific bit assignments T.B.D.. Any PCI transfer which is in progress when a TRAILER word is transferred will be terminated using a Disconnect.

The timing and electrical characteristics of the FIFO interface should be compatible with an IDT72V3690 synchronous FIFO.
3.2 Download/Monitoring Interface

**DATA[31..0]**  Shared data bus (bidirectional)

**ADR[15..0]**  Address bus. Provides address within a chip’s address space.

**CSn**  Chip Selects (active LOW). One will be provided per chip on STC logic board. Decoding will be fixed in FPGA code but easily changed.

**READ**  Read enable (active HIGH). A high level indicates that the ADR, CSn signals are stable and that data should be output to the data bus.

**WRITE**  Write enable (active HIGH). A rising edge indicates that DATA, ADR, CSn are valid and should be latched.

The download/monitor interface provides a simple address+data interface to all chips on the STC logic board. A CS (chip select) signal is activated when the PCI address falls within a programmed range of addresses assigned to each chip. 15 Address lines and 32 data lines are bussed to all chips on the board along with READ and WRITE enable signals. To simplify the electrical design, this interface is “slow”, that is PCI wait states are used to reduce the maximum transfer rate so that set-up and hold time requirements are easily met with the expected large signal fan-out.
The address presented on ADR[15..0] will increment automatically when a PCI bus block write or read is performed (except when PCI AD23 is set as noted below). The CS signals are controlled by the PCI address as follows:

- **PCI AD[17..2]** drive ADR[15..0] on monitoring interface.
- **PCI AD[22..18]** select CS0 through CS31.
- **PCI AD[23]** when high, CS0 is forced on (for Road LUT access). Also, ADR[15..0] will not increment during PCI block transfers.

So, CS0 should be used by the control logic, and CS1..CS8 should be used by the channels. Preliminary timing diagrams for read and write cycles are shown in Figures 4 and 5.

The read cycle begins with Address and CSn asserted. A (programmable) number of PCI “wait states” are inserted to allow for settling of these signals. READ is then asserted to enable the output of data from the selected chip. The data is latched on the trailing edge of READ. One additional wait state is provided for bus disable before the data is transferred to the PCI bus.

The write cycle begins with Address, CSn and Data asserted. WRITE is then asserted after a programmable number of wait states. It is intended that the data be latched on the leading edge of WRITE.

![Figure 4: Download/Monitor READ Cycle Timing](image)

4 Implementation

It is currently anticipated that each of the three interfaces will be implemented in an Altera ACEX 1K series FPGA device, probably in a 256 pin fineline BGA. The EP1K30FC256 or EP1K50FC256 are both currently available and are quite inexpensive.
Figure 5: Download/Monitor WRITE Cycle Timing