



XAPP635 (v1.1) February 23, 2005

Interfacing Virtex-II Series FPGAs With Analog Devices TigerSHARC TS20x DSPs via LVDS Link Ports

Author: Nick Sawyer

Summary

This application note describes a transmitter module and a receiver module compatible with Analog Devices TigerSHARC TS20x digital signal processors (DSPs). These two macros allow double data-rate (DDR) communication of 128-bit words over a four-bit LVDS link at speeds up to 900 Mb/s per line (450 MB/s) when a Virtex-II Pro™ grade -7 device is transmitting, and up to 500 Mb/s per line when a Virtex-II Pro grade -7 device is receiving .

Introduction

Analog Devices TigerSHARC TS20x DSPs contain an LVDS-based high-speed link port function, which is intended for communication between DSPs, but can also be used to communicate with Virtex-II series FPGAs. Obviously, this functionality is extremely useful with systems designed around these two components.

The macros described here are a transmitter module and a receiver module, compatible with the TigerSHARC TS20x DSPs. These two macros allow DDR communication of 128-bit words over a four-bit LVDS link at speeds up to 900 Mb/s per line (450 MB/s) when a Virtex-II Pro grade -7 device is transmitting, and up to 500 Mb/s per line when a Virtex-II Pro grade -7 device is receiving.

Full details on TigerSHARC DSP link port characteristics are available from Analog Devices at: <http://www.analog.com>. The macros described here are intended for use in systems where the clock rate is the same or very close to the same (within 10,000 ppm) in the devices used at each end of the link.

Circuit Description

The logic for the transmitter and receiver functions is straightforward, yet capable of running at very high speeds. A simplified version of the protocol used by TigerSHARC TS20x DSPs is shown in [Figure 1](#). The main thing to note is that the start of frame is indicated by the first rising transition on the clock line. The clock is therefore discontinuous, and so Digital Clock Managers(DCMs) in Virtex-II series devices cannot be used.

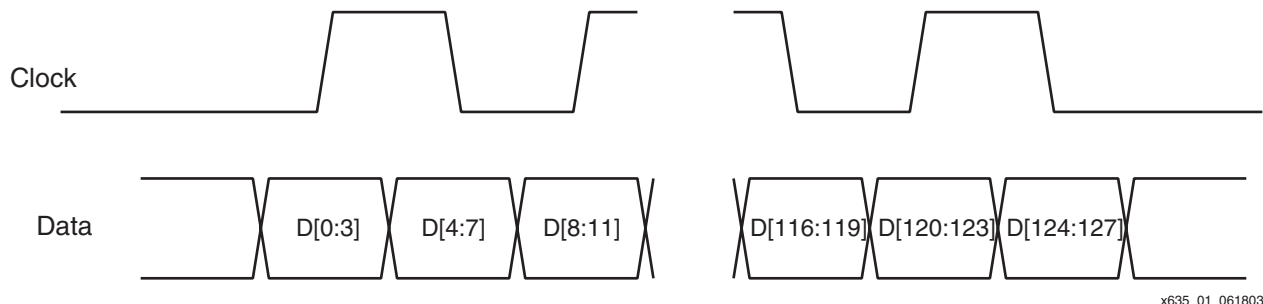


Figure 1: Link Port Function Overview

© 2003-2005 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

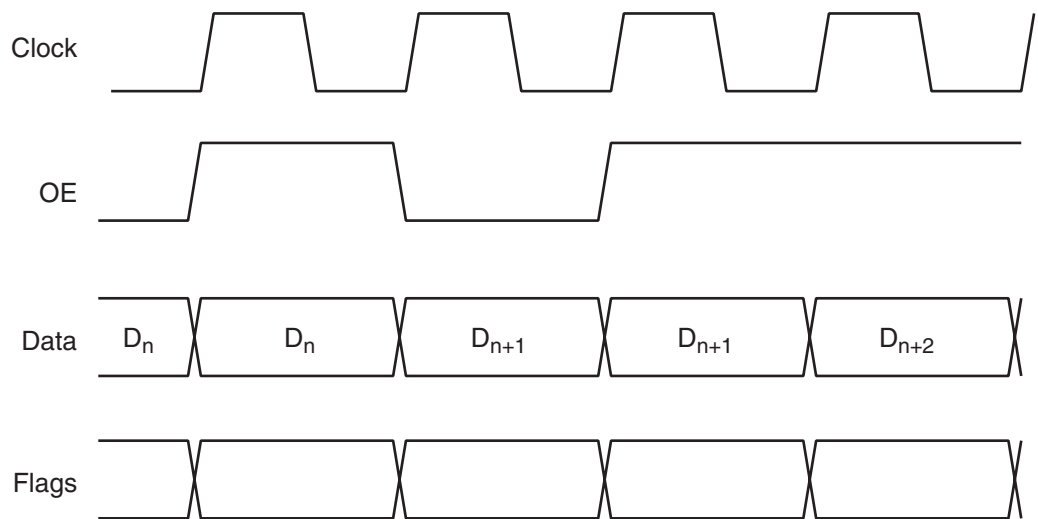
Receiver Macro

Incoming four-bit data is initially clocked into a small dual-port FIFO using a local clock arrangement. The macro has been heavily floorplanned to allow very low skew on this local clock. The placement of the clock and data input pads is critical to operation at high speeds. Further details on this placement are given in the downloadable [xapp635.zip](#) file set).

Four data bits are clocked in by the rising edge of the incoming clock, and four bits by the falling edge. This data is almost immediately clocked out of the FIFO eight-bits wide, synchronous to a continuous FPGA clock that is nominally the same frequency as the incoming link clock. This byte data is written to a larger block-RAM-based FIFO, which is then made available to a 128-bit wide system with data valid timing, as shown in [Figure 2](#).

The translation from eight-bit wide to 128-bit wide data takes place within the block RAM itself. Four memories are used, each of which has an eight-bit input bus and a 32-bit output bus. The depth of the FIFO can be changed easily, depending on latency requirements, and data can be received either word-by-word or in a continuous stream.

One characteristic of the link port protocol is that it can be used with or without the transmission of a verification byte. This version of this application note does not include the logic to allow use of this verification function.



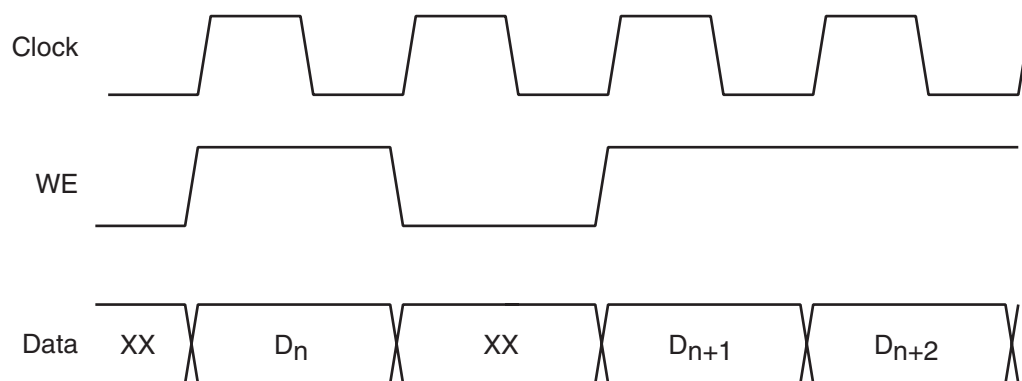
FIFO flags:
 bit0 = empty to 1/2 full,
 bit1 = 1/4 full to 3/4 full,
 bit2 = 1/2 full to full

x635_02_061903

Figure 2: Receive Macro Timing

Transmitter Macro

Data (128-bit) for transmission is written to the macro using the timing shown in [Figure 3](#). The input path is a block-RAM-based FIFO that generates a back-pressure signal (TxBP) when it gets full. Continuing to write data when the back-pressure signal is high can cause loss of data. The macro then internally serializes the 128-bit data to the 4-bit plus clock protocol required by TigerSHARC TS20x DSPs. This is done using techniques similar to those described in [XAPP265](#). Data is transmitted only when the receiver DSP indicates that it is ready by raising its LbACKO line. If this line is asserted Low for long enough, the transmitter FIFO fills up and sends a back-pressure signal to the user logic.



x635_03_061203

Figure 3: Transmitter Macro Timing

Quite a few clocks are required by the transmitter macro. These include:

- The transmitter clock and its complement, and the transmitter clock divided by two, provided by a DCM and three global buffers
- The transmitter clock shifted by 90 degrees and its complement, again provided by a DCM and two global buffers
- The system clock (that is, the internal FPGA clock that generates the data for transmission) with which data is synchronously written

Clock resources can obviously be shared amongst multiple versions of the transmitter macro.

Design Files

The macro design files described in this document are written for use with Virtex-II and Virtex-II Pro devices and are available in both Verilog and VHDL from the Xilinx website at: <http://www.xilinx.com/bvdocs/appnotes/xapp635.zip>. View the enclosed `readme.txt` file for the latest details.

Conclusion

The macros described in this document enable Virtex-II series devices to interface easily with Analog Devices TigerSHARC devices via the embedded LVDS link port function at speeds of up to 900 Mb/s per line (450 MB/s).

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/23/03	1.0	Initial Xilinx release.
02/23/05	1.1	Revised maximum speed.