Version for EDK 8.1i as of May 10 2006

Goals

- To learn about the Xilinx Fast Simplex Link (FSL) Interface
- To become familiar with the documentation you will need to refer to when designing your own systems with FSLs.

Introduction

This module will differ from the more step-by-step modules, as there is a Xilinx application that covers how to connect user IP to the Microblaze FSL bus system. This module is more of a guide through that example.

Background

The FSL is a uni-directional point-to-point connection between 2 devices that Xilinx considers to be a bus. It is essentially an abstracted, glorified FIFO with a standardized interface. The implementation differs depending on if the user needs it to be synchronous (both devices run on the same clock) or asynchronous (the master and slave run on different clocks), on the size, and on some other parameters that the user can specify.
A single FSL can connect only 1 master device to 1 slave device. The master writes into the FSL and the slave reads from it. Since we typically want to be able to communicate in both directions, FSLs are often used in pairs where each of the 2 connected devices is a master on one of the FSLs and a slave on the other.

Requirements

- Access to EDK 6.3i, and the Xilinx Multimedia development board.
- Modules 1-6 so that you have a good working knowledge of EDK, pcores, and ISE.

Preparation

- Read the FSL datasheet:

  `<EDK-installation-dir>/hw/XilinxProcessorIPLib/pcores/fsl_v20_v2_00_a/doc/fsl_v20.pdf`

- Read the FSL Application Note (XAPP529) at

  `http://www.xilinx.com/bvdocs/appnotes/xapp529.pdf`.

  If you are short on time, start reading at the middle of page 6 where they begin to describe the FSL interface in detail.

Guiding You Through XAPP529

- Download the file `m10.zip` into your local directory and unzip it. Make sure this path has no spaces in it! Note that this design is based on the reference design for XAPP529 at

  `www.xilinx.com/bvdocs/appnotes/xapp529_6_2.zip`.

It has been modified to operate with the Xilinx Multimedia Boards.

- Open the project and go to the System Assembly View. Notice that EDK has included the XAPP529 core xil_idct but not the FSLs. The FSLs are considered to be buses in EDK.
- Go to Bus Interface. This is where you add instances of the FSL and connect the devices to the FSLs. Do not use fsl_v20_v1_00_b - it only includes the synchronous SRL_FIFO version of the FSL, which can only be exactly 32 bits wide and 16 words deep. The fsl_v20_v2_00_a version includes the SRL_FIFO version, but also can be used in with other depths, other physical resources, and asynchronously if needed.
- You can see the FSL bus interfaces for the XAPP529 core and 2 pairs of FSL bus interfaces for Microblaze. You connect them to the FSLs here.
- Right click the microblaze_0 instance and select Configure IP.... The parameter C_FSL_LINKS dictates how many pairs of FSL connections are available in Bus Interface. In this design, it is set to 2 even though only 1 FSL link is needed. If you select one of the FSL instances, you can set its parameters. The XAPP529 design uses the older version of the FSL so fewer parameters are available to customize.

Important Notes

- Note that you can only have 1 master and 1 slave on a single FSL.
- Note that 1 FSL link is composed of 1 pair of FSLs. If the FSL link connects devices A and B, one of the FSLs has device A as the master and B as the slave. The other FSL has B as the master and A as the slave.
- Do not write to the FSL when the Full signal is high. Doing so will cause the FSL data to be corrupted. In the asynchronous FSL, doing so will overwrite the word in FSL_S_Data with the latest data word. This is not a concern when doing writes in C code, because the MicroBlaze functions take care of checking the FSL Full and Exists lines.

Step-by-step

- Generate the bitstream and download it to the board.
- Generate the executable for the application file and download it to the board using XMD.
- Connect to the serial port using a terminal program with the following settings:
  Baud Rate: 9600
  Data Bits: 8
  Parity Bits: None
  Stop Bits: 1
  Flow Control: Off
- Run the program (`go`) and observe the output. You should see the results of a sample application which tests the IDCT operation on 8 sets of data. Read through the sample code (`code/system.c`) to see how the software interface to the FSL links works.