Lab: UART-to-SPI bridge

1 Introduction

The general goal of this laboratory is to develop a UART-to-SPI bridge, which takes a UART message of an arbitrary number of bytes and transforms it to an SPI frame. The UART message will consist of multiple UART frames, all one byte in width. The bridge will receive a number of succeeding UART frames, merge them into one SPI frame and output it through a GPIO interface. In order to distinquish seperate UART messages, a new message starts with start byte 0x55. Each message is concluded with a stop byte 0xAA. As a target board the Digilent Basys 3 board is suggested. The SPI communication should take place at 8 Mbps.

2 UART interface

The Digilent Basys 3 board contains a USB-UART bridge that makes the board appear to a PC as a general serial port device. This interface (micro-USB connector) will be used to receive byte frames from the PC. On the PC-side any terminal application can be used to transmit packages (or you could write your own script). Develop and implement a VHDL driver that is able to at least receive data through the USB-UART interface. The baudrate, presence of a parity bit, etc. can be chosen freely. The data payload should be exactly one byte, however. Verify your design before moving to the next steps.

3 SPI interface

Since the received serial data needs to be transmitted through an SPI interface, an SPI driver is necessary. The SPI driver should drive GPIO pins of the FPGA. The start and stop bytes, which are part of the UART message, should not be included in the SPI payload. Use SPI mode 0. Implement and verify your driver, before moving to the next step.

4 Frame merger

UART-to-SPI conversion and transmission is performed after a full UART message has been received. This implies that not every reception of a UART frame needs to be passed to the SPI driver directly. Some sort of controller will be needed to act on incoming UART, keep the status of the number of received packets, merge the received package and control the SPI driver. Design and verify this controller.

5 Final composition

Merge all developed parts to obtain the general objective of this lab. Implement on the target and verify the functionality of your design.

6 Objectives

- Design and verify a UART receiver.
- Design and verify a SPI transmitter.
- Design and verify a system controller and packet merger.
- Glue all sub-designs togeteher. Implement and verify.