Development of generic and portable SPI master/ slave modules in VHDL

Degree programme: BSc in Electrical- and Communication Engineering | Specialisation: Embedded Systems
Thesis advisor: Dr. Torsten Mähne
Expert: Dr. Milos Stanisavljevic (IBM Zurich Research Laboratory)
External project partner: ETEL S.A., Môtiers

In modern Designs, where a lot of embedded peripheral have to communicate with each other, communication links such as the Serial Peripheral Interface (SPI) are very Important. There are a lot of freely available open source implementations. One main restriction, they all have in common, is that they aren't flexible enough to communicate with different slaves. The target of this project is to develop such a highly flexible SPI core, which is fully configurable during runtime.

Introduction

The Serial Peripheral Interface is a widely used and well-known standard for communication in embedded systems. The main problem of the SPI is that the standard isn't exact. Because of this, each of the many SPI slaves on the market may differ in terms of used clock rate, polarization, phase, data size, and timing delays (Figure 2). In addition to this, there is no error detection included in a standard SPI. To address these needs, the company ETEL from Môtiers asked BFH to develop such a configurable SPI core. The present Bachelor Thesis analyses the requirements from ETEL and implements a fully configurable SPI core in VHDL.

Implementation

Based on the requirements from ETEL, the specification of the interface and functionalities of the SPI core were developed and refined to a suitable architecture, which has been implemented using VHDL. The SPI core includes a transmit and a receive path for the communication. It can be configured as a master or as a slave. The rest of the system interacts with the SPI core via register access implemented in its interface block. The control block initiates the communication and the master and slave blocks handle the generation of the serial clock and the slave select signals according to the mode set in the respective cores. To make the core even more flexible, the interface to the processing unit can be configured in terms of data bus with. Common processing systems differ from each other in term of the size of their registers as well as the order in which they save data to them. To adjust to these details, the SPI core can be configured to interface with 8, 16, and 32-bit processors with little Endian or big Endian. The core itself can serialize the data with the least significant bit (LSB) or the most significant bit (MSB) first. (Figure 1)

Fabian Knuchel

Results

The SPI core has been implemented with the required features. The possible minimum and the maximum serial clock rate is determined by the internal clock used in the FPGA. The maximum possible checksum length for the CRC is 32 bits. All components were verified through simulation with automated error injection and monitors. As a demonstration of the SPI core, a loopback system has been implemented, which allows to interactively parameterize the cores from a PC and sending/receiving message frames via a USB-UART interface.

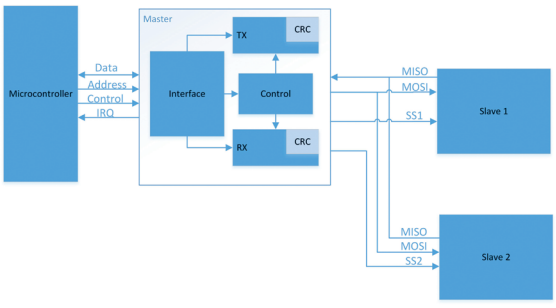

Figure 1: Block Diagram of the Core with the Microcontroller and two Slaves

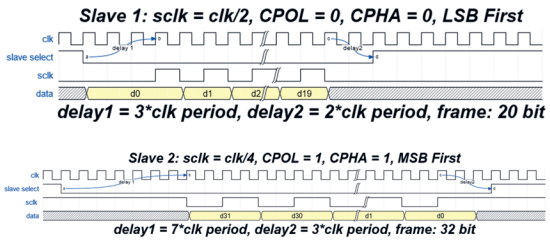

Figure 2: Two Different Slave Communications