# Creating a Low Power Serial Peripheral Interface

## How to implement communications between microprocessors and peripherals, using the Xilinx CoolRunner XPLA3 CPLDs.

by Anita Schreiber
Staff Applications Engineer, Xilinx
anita.schreiber@xilinx.com

The CoolRunner implementation of a Serial Peripheral Interface (SPI) Master described here can be used to add an SPI controller to microprocessors or microcontrollers that do not provide this interface. It will permit direct inter-processor communication and communication with numerous commercially available peripherals.

### Serial Peripheral Interface Protocol

SPI is a full-duplex, synchronous, serial data link. A single SPI device is configured as a master; all other SPI devices on the SPI bus are configured as slaves.

The SPI bus consists of four wires:

- Serial Clock (SCK) - Driven by the SPI master and regulates the flow of data bits. The SPI specification allows a selection of clock polarity and a choice of two fundamentally different clocking protocols on an 8-bit oriented data transfer.
- Master Out Slave In (MOSI) - Data output from the SPI Master and input to the SPI Slaves.
- Master In Slave Out (MISO) - Data input to the SPI Master and output from the selected SPI Slave. Only one selected slave device can drive data out from its MISO pin.
- Slave Select (SS) - Selects a particular slave via hardware control. Slave devices that are not selected do not interfere with SPI bus activities. The SS control line can be used as an input to the SPI master indicating a multiple-master bus contention (SS_IN). If the SS signal to the master is asserted, it indicates that some other device on the bus is attempting to be a master and address this device as a slave. Assertion of SS automatically disables SPI output drivers in the master device if more than one device attempts to become master.

The SCK, MOSI, and MISO pins of all SPI devices on the SPI bus are connected together in parallel.

### CoolRunner SPI Master Implementation

This SPI master design supports the following features:

- Microcontroller interface.
- Multi-master bus contention detection and interrupt.
- Eight external slave selects.
- Four transfer protocols available with selectable clock polarity and clock phase.
- SPI transfer complete interrupt.
- Four different bit rates available for SCK.

A high-level block diagram is shown in Figure 1. The microcontroller (μC) interface is a VHDL module that you can easily modify to support other microcontrollers.

The Address Decode/Bus Interface logic interprets the bus cycles of the microcontroller and performs the read/write operations to the Register File.

The Register File is the interface between the μC and the SPI master logic, and allows the μC to configure and control the operation of the SPI master. Status of the current transfer is provided to the μC via a status register in the Register File. Registers are also included to contain the μC data to be transmitted on the SPI bus and data received from the SPI bus. The SPI Control State Machine controls the shifting and loading of SPI data in the SPI shift registers, and the generation of the slave select signals. The SCK clock logic generates an internal SCK based on the settings in the control register for clock phase, division, and polarity.

### Conclusion

CoolRunner CPLDs operate at the lowest standby power (<100μA) of any CPLD available today, and they are an ideal programmable logic solution for providing interface controllers in portable or power sensitive applications. See www.xilinx.com/apps/epld.htm#CoolRunner for an SPI reference design which contains a detailed application note (XAPP348), VHDL source code, and VHDL testbenches.

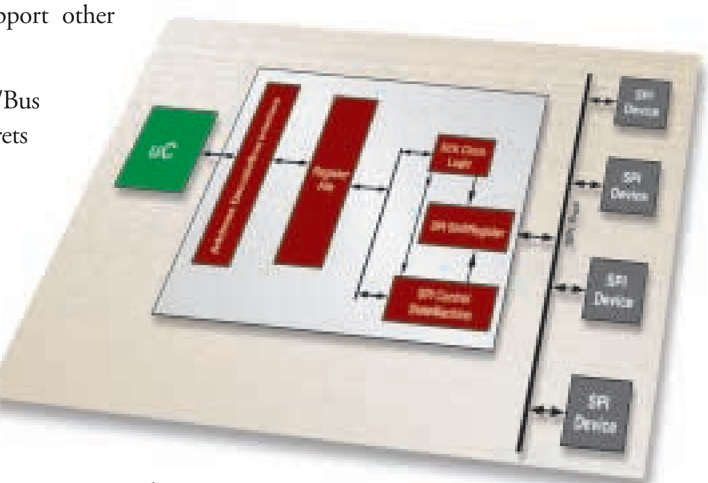

*Figure 1 - CoolRunner SPI master*