

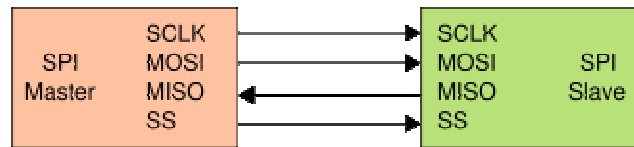
# EE3623 Embedded System Design Laboratory

Dennis Silage, PhD  
silage@temple.edu



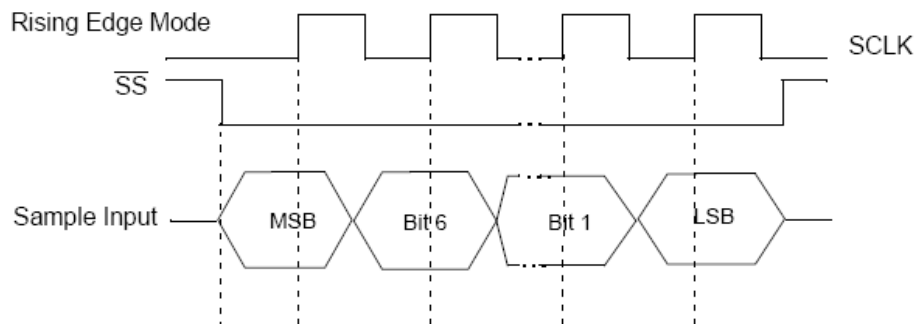
## Serial Peripheral Interface Behavioral Synthesis in Verilog

In this Laboratory you will *design* and *verify* a serial peripheral interface (SPI) sequential logic circuit transmitter and receiver using the Verilog behavioral synthesis language and the controller and datapath construct. The SPI (pronounced *spy*) bus is used to interconnect peripherals on an embedded system. The SPI *master-slave* configuration is shown below.



The SPI bus specifies four logic signals: SCLK Serial Clock (output from master), MOSI Master Output, Slave Input (output from master), MISO Master Input, Slave Output (output from slave) and SS — Slave Select (active low indicates output from master).

The SPI bus uses the active low SS signal to indicate that transmission is to begin on the next rising edge of the SCLK signal, as shown below. The data is transmitted MSB first until the SS signal become inactive high.



You are to implement the following project on the Spartan-3E Starter Board as a behavioral synthesis in the Verilog HDL:

1. Use the four rightmost slide switches for a 4-bit data input.

2. The rightmost 4 LEDs display the data of the slave and the leftmost 4 LEDs display the data of the master.
3. Use push button 0 for the control signal input to the controller of the SPI master to read the 4-bit data in the slide switches, display the data in the leftmost 4 LEDs and to begin the output of data to the slave using the datapath. The slave displays the received data in the rightmost 4 LEDs.
4. Use push button 1 for the control signal input to the controller of the SPI slave to read the 4-bit data in the slide switches, display the data in the rightmost 4 LEDs and to begin the output to the master using the datapath. The master displays the received data in the leftmost 4 LEDs.
5. Use a derived 1 Hz clock from the 50 MHz master clock for the SCLK signal. Note that the SPI protocol indicates that SCLK should be gated and not be running all the time.
6. Use pushbutton 2 as a CLEAR signal for the master and slave datapaths to clear their respective LEDs.
7. The master and slave SPI devices each should be configured as a controller and datapath utilizing finite state machines for sequential operation. Fully describe the configuration in your Laboratory Report.

*Notes:* The EDPGA text is a reference for the Spartan-3E Starter Board hardware and Xilinx ISE EDA.

The clock divider module `clock.v` is described in the EDPGA text on page 103.

The finite state machine and controller and datapath construct in described in the EDPGA text on pages 25-32.

You may want to *debounce* push buttons 0 and 1 for the smooth operation of the SPI master and slave protocol. The push button debounce Verilog module `pbdebounce.v` is described in the EDPGA text on pages 113-114.

Verify that the data transmission is correct (the MSB transmitted is received as the MSB) with several different binary values. Demonstrate the operation to the Laboratory Assistant.

