Serial Interfaces

Introduction

In this lab you will design and test a parallel-to-serial interface that operates similar to the SPI interface's MOSI output.

You'll use a supplied testbench to test your design using test vectors matching the last three digits of your student number.

Requirements

Your design will transmit 4-bit words in order to limit the length of the simulation.

The VHDL entity declaration for your design should be as follows:

```
entity lab8 is
  port (
    data : in std_logic_vector (3 downto 0) ;
    load, sclk : in std_logic ;
    mosi, ss_n : out std_logic ) ;
end ;
```

where the signals are:

`data` the 4-bit parallel data input

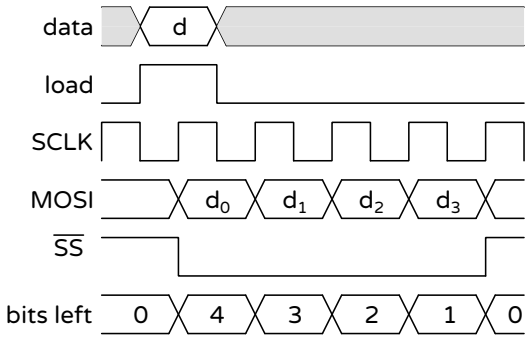
`load` an active-high input that indicates a new value is present on the **`data`** input, this will be active for one clock period;

`sclk` the input clock for your design and also the output clock signal for the serial interface;

`mosi` the master-out-serial-in (MOSI) serial output of your interface, it should transmit the four data bits in order from least-significant to most-significant bit;

`ss_n` the slave select signal output. This should be low while data is being transmitted. The receiver will capture the bits present on **`mosi`** on every falling edge of the clock for which **`ss_n`** is asserted. **`ss_n`** should be asserted for four clock cycles for every value transmitted.

The waveforms for your design should look as follows:



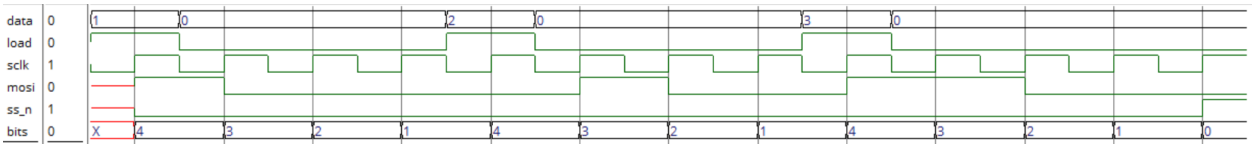
Note that for this design the inputs (**`data`** and **`load`**) change on the falling edge of **`sclk`** but the outputs (**`mosi`** and **`ss_n`**) change on the rising edge. This means that the receiver needs to sample it's inputs on the *falling* edge of **`sclk`**[1]. This also requires **`sclk`** to have an accurately-controlled duty cycle of about 50%. The testbench will generate such a clock for you.

Your design will include the parallel-to-serial converter and a state machine that controls the operation of the interface. It has the following components:

- the transmit shift register is loaded with the 4-bit value on the **`data`** input if **`load`** is asserted otherwise it is loaded with the transmit register contents shifted right by one bit,
- a counter indicating the number of bits left to send is set equal to 4 if **`load`** is asserted, otherwise it is set to one less than the current value if the count is great than zero, otherwise it remains at zero, and
- the **`ss_n`** output is asserted if the number of bits left to send is greater than zero.

Both the **`mosi`** (least-significant bit of the transmit register) and **`ss_n`** outputs should be registered.

[1] Most interfaces can be configured to use either the falling or rising edge of **`sclk`**.



Procedure

Design

Draw a block diagram that meets the above requirements. Write a VHDL architecture corresponding to your diagram. Your VHDL should follow the course VHDL coding conventions.

Simulate

Download the testbench, **`lab8_tb.vhd`** from the course web site.

Follow the procedure in Appendix A of the previous lab to create a simulation project, add the **`lab8.vhd`** and **`lab8_tb.vhd`** files to the project and compile them. After fixing any errors, run the simulation. The Transcript window should show the messages generated by the testbench with the values received over the serial interface and the Wave window should show the signal waveforms.

The waveforms for the default test bench should look as shown above.

Once your design is working, modify the testbench by substituting the last three digits of your student number for the supplied values (1, 2, 3) and run the simulation again.

Collect the files required to demonstrate completion of the lab and submit them as described below.

Submit Results

Submit the following files to the Lab 8 Assignment folder:

- your block diagram. Take a photo and submit the **`.jpg`** file or scan it using a scanner or phone app such as Genius Scan or Adobe Scan and submit the **`.pdf`** file. Make sure your diagram is readable.

- your **`lab8.vhd`** VHDL file

- a screen capture of the waveforms similar to that shown above for the last three digits of *your* student number,

- a screen capture of the Transcript window showing the messages generated from running the simulation similar to that shown below (but for *your* student number):

```
** Warning: NUMERIC_STD.">": metavalue detected, returning FALSE
   Time: 0 ps  Iteration: 0  Instance: /lab8_tb/dut
** Warning: NUMERIC_STD.">": metavalue detected, returning FALSE
   Time: 0 ps  Iteration: 0  Instance: /lab8_tb/dut
** Note: received 1
   Time: 80 ns  Iteration: 1  Instance: /lab8_tb
** Note: received 2
   Time: 160 ns  Iteration: 1  Instance: /lab8_tb
** Note: received 3
   Time: 240 ns  Iteration: 1  Instance: /lab8_tb

SIM 21>
```