ECEN3763 – Lab Week 3

Fall, 2021

Due Friday February 4th, 11:59pm - 35 Points

Purpose:

You will add In-System Source and Probe (ISSP) cores to your Lab Week 1 project in order to enable remote control of your counter. To do this, you will write a short Tcl script. You can use the provided issp.tcl file as a starting point. You will also add a synchronous reset block to your design (reset.sv is provided).

Sometimes you may need to simulate signals in a design to help debug. Depending on the situation, your design may not be connected to the entire system, or perhaps certain inputs are not available (perhaps the front panel has not yet been fabricated). Another scenario where remote access to hardware is useful is when you are not in the same physical location as your board – maybe you are at home, in another lab, or your hardware is in another city, state, or country. As long as a computer is connected to your board via the USB cable, and that computer has Quartus installed, you can connect remotely and issue read and write commands.

The basis for all this remote control is Tcl.

Lab Instructions:

1. Create a new Quartus project for Week 3. You will use the same SystemVerilog source files from Week 1.
2. Add the reset.sv block to your design. This block provides an active low reset signal that deasserts synchronously with the 50Mhz clock.
3. Create an ISSP block to control the 3 pushbuttons. This block should have 3 sources (one for each KEY), and also 3 probes to allow the KEY values to be read by the Tcl script.
4. Create an ISSP block to control the 10 switches. This block should have 10 sources (one for each SW), and also 10 probes to allow the SW values to be read by the Tcl script.
5. Connect the ISSP blocks in your design.
6. Create a Tcl script that allows you to control your counter from a Tcl console.

Submission and Grading:

Your submission will be a single .zip (or .7z or .tar) file uploaded to Canvas for Lab Week 1. The submitted file should contain the following:

a) A Quartus archive of your project (.qar file)
b) Your Tcl script
c) You may choose to include a brief text file describing issues with your final design (if

any), an explanation of your simulation (if needed), and anything else you want me to know about your submission.

Lab Week 2 must be a separate submission from Lab Week 1.

This lab will be graded as follows:

Quartus project that successfully compiles and downloads to board:	5 points
Counter design operating on your board:	15 points
Functional Tcl script to control your design:	15 points

Additional Guidance for this Lab:

1. As noted in the reset.sv file, you must connect the pll_locked input of the reset block to logic 1, otherwise your design will be held in reset.
2. When using an ISSP source to emulate a KEY, you should combine the KEY and the ISSP probe with a logic gate. Your counter design should function normally after programming, without an interaction with the ISSP blocks. When you run your ISSP Tcl script, the ISSP sources should then override the behavior of the KEYs and SWs.
3. The sources of an ISSP can have initial output values defined. By default, all sources drive to logic 0 after programming the FPGA.
4. Since Tcl is an interpreted programming language, you can execute individual commands at the Tcl command prompt. Once you know the commands are correct, you can add these commands to your script.
5. Including the Quartus tools in your Windows path will allow the Tcl console to always be found when you start the console. Assuming you used the default installation location for Quartus, you will need to add **C:\intelFPGA\20.1\quartus\bin64** to your path statement.
6. You can run a simulation of your completed design, but I do not know how to manipulate the ISSP blocks from within Modelsim, so your simulation will look the same as the simulation from Week 1.