

Lab Assignment 5: Stop-Watch Design

ECE/CS 3700

Spring 2009

Assigned Tue (March 24) onwards, Due date: your respective lab sessions during the week 4/6-4/10.

I. LABORATORY OBJECTIVES

In this lab, you will design and implement a **stop-watch** on the XESS board. You will follow the Verilog Design, Simulation, Synthesis and Mapping, and FPGA implementation as you followed in the previous classes.

What are the new concepts that you will learn in this lab? First of all, the stop-watch is like a counter, so you will learn how to design counters. Secondly, you will learn “how to count time” using a reference clock. Thirdly, so far we have used Verilog to design only combinational logic using gate instantiations, assign statements or using the always (behavioural) constructs. For this lab, you will learn how to write “behavioural” Verilog code to model “sequential” behaviour. Moreover, how would you simulate a sequential circuit? How would your testbench generate a clock? So, these concepts are new. Enjoy!

II. DESIGN SPECIFICATIONS

Your design will be a single (decimal) digit stop-watch with *start/resume*, *stop*, *reset* buttons. The functions for each of the input buttons are as follows:

- **Start/Resume** - This button is used to trigger the counting on the stop-watch. When this input is activated from an initial state, **the watch advances every second** counting from 0 to 9 and then looping back to 0. The counting process should just keep on running until and unless any other input (stop or reset) is activated. If this input is activated from a previous steady display on the watch, then the watch resumes counting every second from that particular display value.
- **Stop** - This button is used to stop the stopwatch counting and to hold the display. When this button is pressed the stopwatch stops counting at the very instance and displays the time (in seconds) continuously, until any other input (resume/reset) is pressed.
- **Reset** - This button is used to reset the stopwatch display to 0. When this button is pressed, irrespective of the state of your counter, your stop watch is reset to zero (the display should read 0) and should hold this value zero until the *start/resume* button is activated.

A. Input-Output Interface

Use the push buttons PUSHB2, PUSHB3, PUSHB4 (corresponding to s3, s4, s5) available on the XST board as the inputs and the 7-segment display LED on the board to display the output. You are to program the FPGA on the XSA board with the necessary code to achieve the above stopwatch design.

How to Track Time? Use the internal clock on the XSA board as the reference for computing time. *Hint: If I have a clock running at 60 MHZ, how many positive edge triggers will I observe in 1 second?* The intellectual challenge here is to do the necessary “clock division” within your Verilog code to increment the counter *every second*. You will have to refer to the clocking issues w.r.t. XSTOOLS (refer to 'gxsssetclock' in the XSA-50 users manual).

III. THE ASSIGNMENT

- Design the above stopwatch.
- Write a behavioural Verilog description for the same.
- Write a testbench to simulate your design “exhaustively”.
- Synthesize the design and observe the synthesis, map and place-and-route reports. What new information do you find in these reports *viz-a-viz* combinational synthesis (last lab assignment)?
- Download the above implementation onto your XSA board and demonstrate correct operation of your circuit to the TAs.

Deadlines: This is a two-week long lab. The entire class should demonstrate the final circuit operation to the TAs during your regular lab times, 04/06 - 04/10. Reports are also due by the end of the week (4/10) by 5pm. Drop it in the locker or hand it to the TA.

Report: As usual, your report should have a brief write-up of your objective, computations (if any), your approach, the Verilog Code, observations, results and conclusions. Attach appropriate code, test benches, waveforms, or whatever you deem necessary. Just don't overdo it!

Have fun with this lab. Please do not miss your lab sessions. There are some new concepts, and the TAs will give you a brief description of how to approach the problem and will give you some guidelines as to how to design the Verilog code. If you have any questions, feel free to ask me or the TAs.

Apprécier l'expérience!