CS M152A Lab 3

Stopwatch

In this lab, you will be designing a stopwatch on the FPGA.

Introduction

In this assignment, you will go through the complete FPGA design flow by designing a stopwatch circuit and implementing it on the Nexys™3 Spartan-6 FPGA Board. The inputs of the stopwatch are buttons and slider switches. By implementing it you will review the design techniques you learned from previous labs. The digits of the stopwatch are displayed through the on-board seven segment display. To configure it you will need to study and explore the reference manuals, which is also an essential step in more realistic and complex digital design tasks.

Functionality

The stopwatch will start as a basic clock which counts minutes and seconds. The left two digits of the seven-segment displays will count minutes, and the two on the right will count seconds. For example, without touching anything, after 1 minute and 43 seconds, the stopwatch should display: "0143". The stopwatch can be controlled using two user-defined buttons: pause and reset.

The time stored in the stopwatch should be adjustable by the use of the following inputs: SEL[1:0], NUM[7:4], and ADJ. Both of these inputs are switches on the FPGA.

- **SEL** are 2 switches which will choose among the four individual digits to be adjusted in the adjustment mode.

SEL	Selected
11	Tens Digit, Minutes
10	Units Digit, Minutes
01	Tens Digit, Seconds
00	Units Digit, Seconds

- **NUM** are 4 switches that represents binary numbers.

- When the **ADJ** input is set to logic high, the clock is in adjustment mode. During this mode, normal increments of the clock is halted. Instead, the selected digit (selected by **SEL**) of the clock can be set to the value specified by **NUM**. A user should be able press **PAUSE** to set the value. The unselected portion of the clock is effectively frozen while in adjustment mode. In addition, the selected portion of the clock should blink in the adjustment mode.

ADJ	Action
0	Stopwatch behaves normally
1	Stopwatch stops and selected digits is adjustable

In addition to the two switches, your circuit will use two push buttons (on the FPGA) which will control the timer behavior.

- **RESET** will force all of your counters to the initial state 00:00
- **PAUSE** in normal mode will pause the counter when the button is pressed once, and continue the counter if it is pressed again. In adjust mode, pause will set the digits to the corresponding value.

Implementation

Counter

The basic building block of the stopwatch is the decade (modulo 10) counter. The seconds counter will increment every second, and every 60 seconds, it will reach the maximum value of 59 and enable the minutes counter, which will increment on the next rising edge of the clock.

Clock

For this assignment, you will be using four different clocks – a 2 Hz clock, a 1 Hz clock, a much faster clock (50 – 700 Hz), and a clock for blinking in the adjust mode (>1Hz). We recommend that you create a Clock Module that takes the 100 MHz master clock (internally connected to pin V10 of the FPGA board) as input and outputs 4 different clock signals.

The main purpose of the third, faster clock is to assist in the multiplexing of the seven-segment display. The display is designed in such a way that all four seven-segment displays will show the same digit if no additional circuitry it used. For this reason, you will need to cycle through the four digits using a much faster clock so the human eye sees four digits. The cycling frequency should be between 50-700 Hz; we'll leave it to you to find a reasonable value.

In addition to the function mentioned above, the faster clock can also serve as the under sampling clock that filter noises in the debouncer circuit.

The clock for blinking in the adjust mode needs to blink at least 1Hz. It is up to you to choose a blinking rate that looks good. Note that this rate cannot be 2Hz because we will not see the minutes or seconds increment correctly while adjusting.

Seven-Segment Display

Introduction to the seven-segment display on the Nexys3 board can be found on page 18 of the Nexys3 Reference Manual. As required for every FPGA and embedded systems engineer, you are required to read the manual to understand how it operates and how you implement the design. In short, the digits share the same logic for the segments, so they should be enabled one at a time to display different digits. Note that for both the digits logic and the segment logic, 0 is enable.

Debouncers

In previous labs, you've already got familiar the concept of debouncing. In this lab, you will also need to apply debouncing to the buttons. In particular, you should make sure that the *pause* is very responsive in order to enable precise control.

User Constraint Files (UCF)

Implementation constraints are instructions given to the FPGA implementation tools to direct the mapping, placement, timing or other guidelines for the implementation tools to follow while processing an FPGA design. Implementation constraints are generally placed in the UCF file, but may exist in the HDL code, or in a synthesis constraints file. Examples of implementation constraints are LOC (placement) constraints and PERIOD (timing) constraints.

With the help of master UCF file (nexys3.ucf), you should be able to create a UCF file for this project. You can refer to the nexys3 reference manual for further information on the "LOC" constraints.

Deliverables

When you finish, you should demo your design to the TA, and explain your design to him/her. The following should also be submitted for this lab:

1. Project Code: your code should be uploaded to the corresponding assignment page on the course website
2. Lab Report (Electronic Version): the lab report should be uploaded in the corresponding assignment page
3. Lab Report (Paper Version): the paper version of the lab report should be printed out on both sides and handed in on assigned date