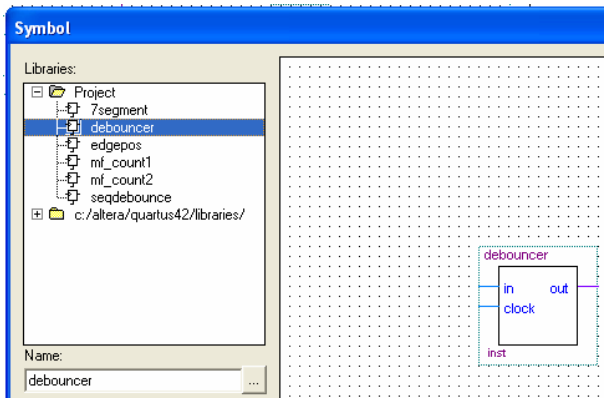
FPGA Lab 4 – Debouncer Circuit

Purpose: In this lab we will build a digital debouncing circuit that will produce a single clock-cycle pulse for each push of the pushbutton switch on the LiveDesign board. This handy circuit will be used in future labs where we will use these pushbuttons to control logic. This design will be the first time that you will build a state machine, which we will use often in the rest of the course. In addition, you will learn how the Quartus software makes modules out of your code so that you can make more flexible and modular designs.

1) **Lab4.1 : Push button counter.** In this lab you will use a counter to measure the bounce coming from the push-button switch. Copy the Lab4.1 to your folder and open the project. In this folder you will see the push button inputs and the 7-segment outputs. Design a circuit to: a) synchronize the push-button logic to the clock, b) produce a single pulse for each rising edge of the push-button signal (using the edge detector from the homework and class), c) count the pulses, and d) display the count.
2) By testing the circuit with many pushes of the button, make a rough estimate of the probability and number of bounces coming from your switch.
3) **Lab4.2 : Debouncer Circuit.** Design a circuit to debounce the push-button input using the assumption that after a transition, the logic level should stay at that transition value for at least a fixed amount of time. Design a state machine following the discussion of class. Test the debouncing circuit for a delay ranging for 2^10 to 2^20 clock cycles. What time delay is needed in order to ensure no bouncing of the switch? (Note: you can use a time delay up to about 0.1 s before affecting the functionality of the switch since it would be hard to push the button more than a few times per second). Much of the code for the state machine has already been entered in the text module "seqdebounce". Open the module and edit the truth table to complete the state machine. Text entry with a truth table has been chosen for defining the state machine because this is the simplest method to enter all the possible combinations for the flow of states. Please note how the input and outputs are defined, as well as the state machine variables, since in the future you will be designing from scratch state machines.
4) In this class our state machines will be defined ("encoded") according to your design. Although it's possible for Quartus to automatically encode them, we want to make sure this feature is disabled in the software so as to not override your design. Select "Assignment/Settings…", and then make sure the State Machine Processing is set to "User-Encoded" and NOT to "Auto".
5) Test the debouncing circuit for several time delays, spanning about 2^10 to 2^20 clock cycles. At what time delay do you no longer see switch bounce? This is approximately a few times larger than the time-scale for the bounce events.
6) **Lab4.3 : Modular Design.** Copy your working project of lab4.2 to a folder named lab4.3. In this project you will redesign lab4.2 with the debouncing function and the edge detector being modules.
7) Select "file/new" and choose to make a "block diagram/schematic" file. This creates a new schematic diagram module, which you should rename "debouncer" using the "file/save as" command. Go back to you top level "LiveDesign" module, and select (via left click – drag) the part of the circuit that functions as the debouncer. Right click on a part and select "copy", change screens to the "debouncer" module, and then right click "paste". Using the symbol

tool (AND gate symbol) create input pins “clock” and “in” and an output pin “out”. Connect these pins to the appropriate wires.

8) Now create a block diagram symbol that you will be inserting later in your top-level design. First save the file with “File/save”. Then chose “File/Create.update/Create symbol files for current file” and the design will be precompiled and checked for errors. If compilation errors occur, you have to fix them at this time.



9) Change screens to the top-level “Live Design” module, and insert this module. This is done with the symbol tool, choosing the file “project/debounce”. Insert this into your design and wire the inputs and outputs. Note that if you want to debounce more than one push-button, this is easy to do by simply copying and pasting this module multiple times.
10) Repeat the above module for a positive-edge detector module “edgepos”.
11) Compile and test for proper functioning of the modules. In the next lab you will learn how to import these modules as libraries into other designs.