Problem Statement (Lab 17-02-2016)
Part-I

Design a debouncer logic in the VHDL.

What is debounce logic?
When you press a key on your computer keyboard or any switch, you expect a single contact to be recorded by your computer or application using switch. In fact, when a key/switch is pressed it make an initial contact and then bounce-back or lighten up the contact. A key/switch undergoes multiple rounds of such contact and bounce-back before making a stable contact. Fig.1 shows the switch bounces produced when a switch is pressed.

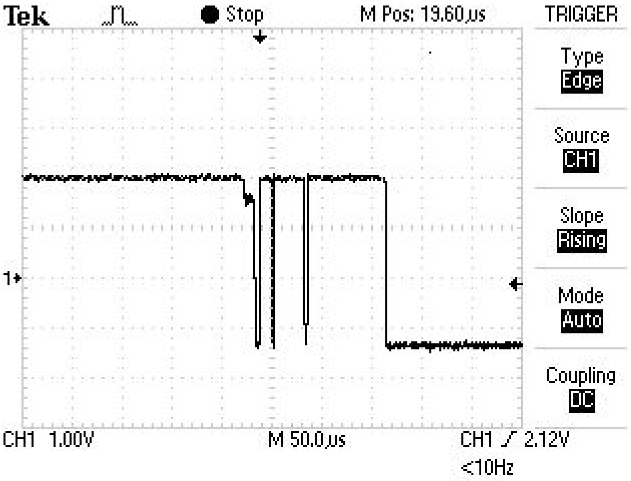

Figure 1 Switch bounces

Switch debouncing is one of the methods employed in digital circuits to ensure a single press doesn't appear like multiple presses.

Design Specification:
You will have following interface ports

Input:

Clk : 1 bit clock

Reset : 1 bit reset

Switch_btn : 1 bit switch button

Output:

Debounce: 1 bit debounce out

Whenever, *switch_btn* value become *high,* wait for 1000 clock cycle to ensure that *switch_btn* is in stable state and it had not become *high* due to any glitch in circuit or

improper contact. As soon *switch_btn* become *low* (zero) you need to reset your clock counter (running to calculate 1000 clock period) to zero.
After 1000 clock cycles if *switch_btn* is still *high*, stop counting and make your *debounce* signal *high* for one clock cycle i.e every time your counter reaches a value of 1000 you need to make *debounce high* for single clock cycle, otherwise it is always 0.

When *reset* signal become *high*. It sets all counter to initial value of zero and makes all outputs to their default value (usually zero).

Part-2

After finishing your design of part-1, now you need to integrate *debouncer logic* with your *stop watch module* completed in last lab.
In *stop-watch module* you had a signal for *start_stop*. Now you will connect your *start_stop* signal of the *stop-watch module* to *debounce* signal coming from *debouncer logic* as shown in fig.2

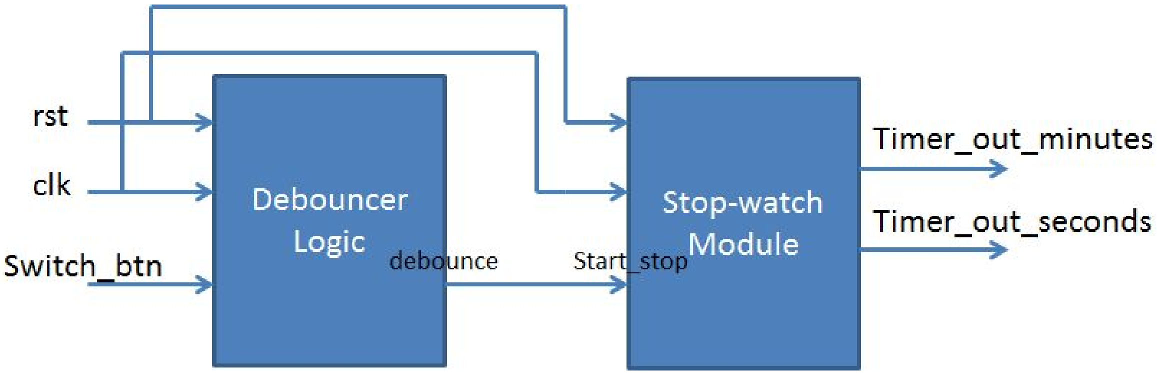

Figure-2 Block level view