HW I

1) Design a 3-bit up/down counter with reset feature. If input x equals to 1, then counter counts from 0 to 7. If x equals to 0 then, counter stops and starts to count downwards. If reset is zero then counter value becomes zero.

    a. Draw its state diagram.
    b. Implement this design by using VHDL. Make its simulation using ISIM simulator.
    c. Show the schematic of your circuit.

2) Design a clock divider that divides clock to 2,4,8,16 according to binary input of x (x equals “00”,”01”,”10”,”11” respectively.). Assume that you have 100 MHz clock frequency.

    a. Draw its truth table.
    b. Implement this design by using VHDL. Make its simulation using ISIM simulator.

3) Design an 8-bit up counter with reset feature. Two different count speed should be defined. One of the count speeds should be 1 bit at 1 sec, while other one should be set 4 bit increment in 1 sec.

    a. Draw its block diagram.
    b. Implement this design by using VHDL. Make its simulation in ISIM simulator.

4) Make an implementation on FPGA given to you (Virtex-5) with the followings.
    a. 4-bit up-down counter which increments and decrements automatically.
    b. 4-bit up-down counter which is incremented and decremented by a switch.
    c. 4-bit up-down counter which is incremented and decremented by a button. (Use a debouncer.)
    d. Add (b) to a start/stop button.

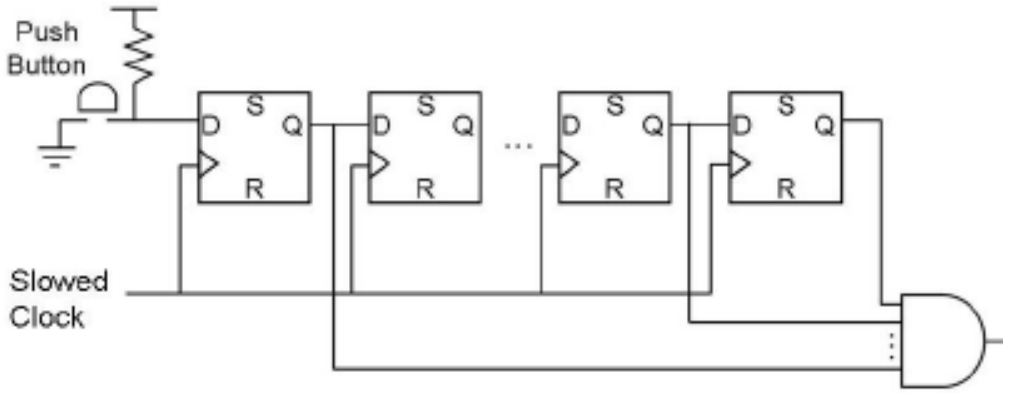
    For all sections of (4), a switch for clock frequency should be arranged. If switch is zero, clock frequency should be 1 Hz and for 1, it should be 100 Hz. All counters should have synchronous reset property. Show the counter results on leds. You can see the details of programing an FPGA in “Programming_FPGA.pdf” file provided to you. In “attribute.txt” file, you can see pin assignment syntax.

Debouncer Circuit:

You should condition the output of the push buttons before you use them as inputs to your system since pressing the button once actually generates many glitches. You can use a shift-register to detect if the button is pressed and then generate only one clean input signal.

Your FPGA has the five push buttons as input ports. After you synthesize the design, before you transfer the code to the FPGA itself, you should map these input signals to the related pins of the FPGA from the user guide.

In order to use the push buttons as inputs, they must be debounced so that a single push of the button results in a clean signal. One way to debounce a switch signal is to shift the bouncing signal down to a shift register. Then the outputs of all the flip-flops can be anded to generate a clean signal. The frequency of the global clock that the FPGA uses, which is generated by the board is 100 MHz. Assuming the bounce interval of a switch has been experimentally determined to be about 3 ms, we should design a clock divider circuit using an N-bit counter that will divide the 100 MHz clock into a clock rate suitable for switch debouncing. Note that an n-bit counter divides the frequency of a clock signal by 2n times. Determine how many flip-flops you need to use. The output of this circuit should be input to your actual circuit.



Deliverables

Your reports (*.doc or *.pdf) should include the snapshots for all simulation results, a summary of the synthesis results (do not simply copy/paste the *.syr file, present the results in tabular form if possible), schematics of the synthesized circuits, and **your comments**. Important considerations that went into the design, and extra features, if any, should also be documented. Your reports should not contain your VHDL codes, which will be submitted as individual files. Before delivering your homework, zip all relevant VHDL files including testbenches (*.vhd), synthesis output files (*.syr), add a bitstream file (*.bit) if a demonstration is made and your report under a folder named as: *hw1_yy-mm-dd_yourfirstname_yourlastname*. Only the following file types should be in the unzipped folder: *.doc (or *.docx or *.pdf), *.vhd,*.syr, *.bit.