CS223 Laboratory Assignment 4

Build Counters using D Flip-Flops

Lab dates and times:
Section 1: 13.10.2017 Monday 08:40-12:25
Section 2: 14.10.2017 Tuesday 08:40-12:25
Section 3: 13.10.2017 Monday 13:40-17:25
Section 4: 14.10.2017 Tuesday 13:40-17:25
Section 5: 16.11.2017 Thursday 08:40-12:25
Section 6: 17.11.2017 Friday 08:40-12:25

Location: EA Z04 (in the EA building, straight ahead past the elevators).
Groups: Each student will do the lab individually. Group size = 1

Preliminary Design Work (30 points)

In Lab4, you will implement two counters: a normal up-counter and a special counter called gray code counter. You will implement both of your designs on the FPGA. SystemVerilog modules should be prepared in advance, and assembled neatly into a Preliminary Report with a printed cover. Each page should have a proper heading. The content of the report is as follows:

a) A cover page which includes the following: course name and code number, the number of the lab, your name and student ID, date, number of your trainer pack.

b) Write a **structural** SystemVerilog module for a 2-bit up-counter which counts the sequence 0,1,2,3,0,1,2,3,... First, you need to define a D Flip-Flop module with synchronous reset and preset inputs (to set the stored bit to 0 and 1, respectively). Then, you need to use it two times to connect all sub-modules according to the logic diagram in Figure 1. Your module should have 1) clock input, 2) synchronous CLEAR input for resetting counter to "00", 3) synchronous PRESET input for setting counter to "11", and 4) a 2-bit output from Flip-Flops corresponding to the counter value. You also need to prepare a testbench for it.

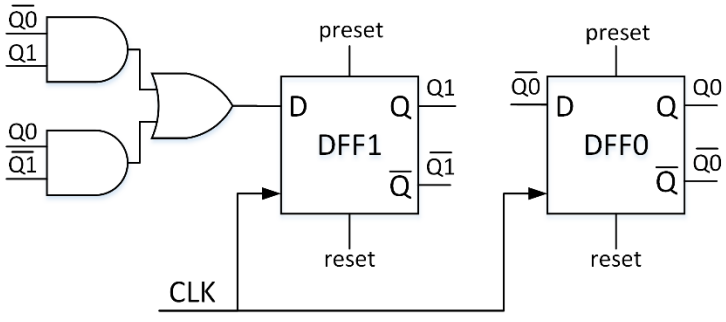

Figure 1: logic diagram of a 2 bit up- counter

c) Using case or if-else statement, write a **behavioral** SystemVerilog module for a 3-bit gray code counter. As illustrated in Figure 2, in this type of counter only one digit changes every time. Your module should have 1) a clock input, 2) synchronous CLEAR input for resetting

counter to “000”, 3) synchronous PRESET input for setting the counter to “111”, and 4) a 3-bit output from Flip-Flops corresponding to the counter value. You also need to prepare a testbench for it.

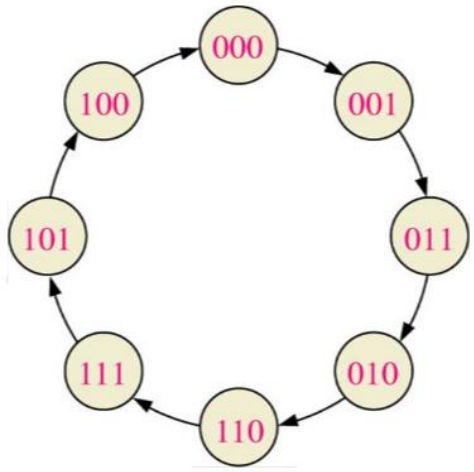

Figure 2: 3-bit gray code counter

The Preliminary Report will be **turned in at the <u>start</u> of the lab**. Therefore, you may need another copy of your designs and SystemVerilog programs with you in the lab to refer to or possibly correct and change it.

Part 1: Simulation (35 points)

In first part, using Vivado tool and the benchmarks that you have already prepared, simulate both of your counters. In simulation, you need to show the requested functionality of all input pins and proper counting order of both counters. When you are convinced that your codes work correctly, show the simulation results to your TA. Be prepared to answer questions that you may be asked. TA will check that the SystemVerilog code of first counter is in structural form and the second in behavioral form.

Part 2: Counters on the FPGA (35 points)

For this part, the standalone BASYS3 board is enough and you do not need to connect it to Beti board. After simulation, using Vivado tool, compile and program your code to FPGA. You need to connect, CLEAR and PRESET inputs to switches, and three FFs as outputs to LEDs. Because the frequency of clock signal of Basys3 board is very high (100Mhz), if you use it directly, you will not see any change on LEDs. By a clock divider module (see below code), we reduce the clock rate from 100Mhz down to around 1Hz, so the counter counts once per second. Instantiate this module in your top module, then feed the original Basys3’s clock input to input of ClockDivider, and use its output in your design. When you are sure that everything is ready, call the TA to show your work.

```
// clk_in is the original clock signal coming from the FPGA board (100 MHz)
// clk_out is the slowed-down clock signal that you can use for your modules (~1Hz)
module ClockDivider(
    input clk_in,
    output clk_out
    );

logic [26:0] count = {27{1'b0}};
logic clk_NoBuf;

always@ (posedge clk_in) begin
        count <= count + 1;
end//always

assign clk_NoBuf = count[26];

BUFG BUFG_inst (
  .I(clk_NoBuf),  // 1-bit input: Clock input
  .O(clk_out) // 1-bit output: Clock output

);
endmodule
```

Basys 3's clock input is available on pin 'W5'. Assuming that the name of the clock input pin in your top module is "clk_in", you need to add below lines into your constraint file:

```
#------------------------------------------------------------
#---------------------- Clock signal ----------------------
#------------------------------------------------------------
set_property PACKAGE_PIN W5 [get_ports clk_in]
set_property IOSTANDARD LVCMOS33 [get_ports clk_in]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk_in]
```

Submit your code for MOSS similarity testing

Finally, when you are done and before leaving the lab, you need to upload the file *StudentID_SVerilog.txt* created in the Implementation with FPGA part. Be sure that the file contains exactly and only the codes which are specifically detailed above. Don't include the codes which are given to you. If you have multiple files, just copy and paste them in order, one after another inside text file. Check the specifications! Even if you didn't finish, or didn't get the SystemVerilog part working, you must submit your code to the Unilica Assignment for similarity checking. Your codes will be compared against all the other codes in all sections of the class, by the MOSS program, to determine how similar it is (as an indication of plagiarism). So be sure that the code you submit is code that you actually wrote yourself! All students must upload their code to the 'Unilica>Assignment' specific for your section. Check submission time and don't miss it before leaving the lab. After taking a backup of your work, don't forget to delete it from computer. Because students of other sections will work with your system too.

Clean Up!

1. Clean up your lab station, and return all the parts, wires, the Beti trainer board, etc. Leave your lab workstation.
2. CONGRATULATIONS! You are finished with Lab #4 and are one step closer to becoming a computer engineer.

NOTES

- Advance work on this lab, and all labs, is strongly suggested.
- Be sure to read and follow the Policies for CS223 labs, posted in Unilica.

LAB POLICIES

1. There are three computers in each row in the lab. Do not use middle computers, unless you are allowed by lab supervisor.
2. You borrow a Lab-board containing the development board, connectors, etc. in the beginning. The lab supervisor takes your signature. When you are done, return it to her, otherwise you will be responsible and lose points.
3. Each Lab-board has a number. You must always use the same trainer board pack throughout the semester.
4. You must be in the lab, working on the lab, from the time lab starts until you finish and leave. (Bathroom and snack breaks are the exception to this rule). Absence from the lab, at any time, is counted as absence from the whole lab that day.
5. No cell phone usage during lab. Tell friends not to call during the lab hours--you are busy learning how digital circuits work!.
6. Internet usage is permitted only to lab-related technical sites. No Facebook, Twitter, email, news, video games, etc--you are busy learning how digital circuits work!.
7. If you come to lab later than 20 minutes, you will lose that session completely.
8. When you are done, DO NOT return IC parts into the IC boxes, where you've taken them first. Just put them inside your lab pack box. Lab coordinator will check and return them later.