CS203: Lab 5 and 6

Implement a combinational/sequential circuit using configurable logic

Objective: Implementing a combinational/sequential circuit using configurable logic

Language: Verilog
Points: 20 each
Release date: 28 Oct 2020
Deadline lab 5: 11 November 2020
Deadline lab 6: 22 November 2020

Detailed problem statement:
Following are possible circuits:

I. 4 bit adder (input two numbers of 4 bits and carry. Output 4 bit sum and carry)
II. BCD adder (input two numbers of 4 bits and carry. Output 4 bit sum and carry. Both input and output is in BCD format)
III. 8 to 1 multiplexer (input 8 data inputs, 3 control inputs, one output)
IV. 3 to 8 line decoder (Input: three bits, output: 8 lines)
V. 8 to 3 encoder (Input: 8 bits. Assume only one of them would be high at one time. Output: three bits)
VI. 4 bit synchronous counter (two bit input is ‘c’ and ‘d’. If ‘c’ is ‘1’ then counter increments, if ‘d’ is 1 counter decrement. Both c and d can not be 1 at the same time. If both are ‘0’ flip-flops holds their values)
VII. 8 bit universal shift register with serial as well as parallel input, parallel output.

Lab 5:
Your circuit id is (last four digits of your entry number) % 7 + 1. Implement your problem id using logic tiles and switch boxes designed in Lab 4.

Lab 6:

In lab 6, you implemented three circuits from the list using “the same” configurable logic. Two circuits in addition to circuit of lab 5 are:
If lab 5 circuit id was I or II : your circuit id for lab 6 are =(last four digits of your entry number) % 3 + 3 and (last four digits of your entry number) % 2 + 6
If lab 5 circuit id was III or IV or V : your circuit id for lab 6 are =(last four digits of your entry number) % 2 + 1 and (last four digits of your entry number) % 2 + 6
If lab 5 circuit id was VI or VII : your circuit id for lab 6 are =(last four digits of your entry number) % 2 + 1 and (last four digits of your entry number) % 3 + 3

“The same” means - FPGA.v should be the same for all the three circuits of lab 6. It could be different from lab 5 design.

Details:

1. Design consist of three files: FPGA.v, configure.mem, and testbench.v
2. In FPGA.v, you can use only logic tile and switch box in your verilog. Verilog code should be structural, with no dataflow or behavior statements. No other logic or operator can be used.
    a. Any number of instances of logic tile and switch box could be there.
    b. Switch boxes and logic tiles can be connected in any ways.
    c. Number of inputs and outputs in FPGA.v would be maximum of inputs required by your circuits and maximum of outputs required by your circuits. Clock will be additional input.
3. configure.mem: Configuration file format can be decided by you.
    a. Ideally, any circuit could implemented just by changing configuration file.
    b. For lab 5, you are suppose to submit configure.mem corrspending to your circuit id
    c. For Lab 6, there should three separate configuration files need to be submitted one for each circuit id assigned to you.
4. testbench.v: Testbench will read the configuration file, and will assign it to your verilog module.
    a. Testbench will have the testcases for testing circuit not the configurable logic.

Logistics:

- Submit lab5.v, configuration file, testbench.v and vcd file.
- Zip the above files into “Lab5_<entry_number>.zip” and submit on moodle. Use only zip for compression, do not use rar, tgz, bgz or any other compression.

Evaluation:

- This lab will be evaluated in-person (using google meet) at the end of course, along with lab 6.
- Each student would get 15-20 minutes time for demo and viva. Vivas would be scheduled on 23 and 24th Nov
- Course policy of late submission and plagiarism, as given on course webpage would apply.

The theory regarding the assignment is covered in the lecture on 12 October 2020.

FAQs

Q: Is it necessary to include a flip flop and a mux in each of my logic tiles as you have shown in the project explanation video, if yes what are they used for?

A: Yes, they are required. Idea to have “same” internal structure for all the logic tiles. This helps in lesser complexity while selecting optimal logic tile for our combinational/sequential logic.

Q. Do we need reset for FF in logic tile?

A: For the counter to work properly, FF should be initialized to 0. It is advisable to have an initial section in the logic tile module, that initialize output to ‘0’.

For register task also, this reset functionality should help.

Q: I wanted to know if I could make a module that performs a certain small task and uses logic tile, and then use that newly defined module in my main module that does all the tasks that are needed for the final task to be completed. For instance if I need a 5 bit bitwise OR then I make a small module that uses logic tile and then use that newly made module in my main module for that small task. This will break make my code into chunks and it will be easier to work with.

In short, the question is: Can we have a hierarchical design in lab 5?

Answer: No, Lab 5 will be a flat design, using only logic tile, and switch boxes.

Q: *If I need to assign memory to logic tile through the test bench or if I can do that directly in my modules that need them.*

A: Similar to lab 4, assigning memory to logic-tile would be done by the testbench.

Q: How lab 5 and lab 6 are different?

A: Lab 5 will act as a pilot project, and will give you an understanding of working with configurable logic. With that experience, you will be able to come up with a more generic design of a configurable fabric that can be configured as multiple applications.

Q: For counter will the inputs c and d keep changing or stay constant? For example, if case c = 1 and d = 0 is given then do we have to run the 4 bit counter for a particular amount of time, or wait for another input?

A: c and d both are received as synchronized input, that is, they are sampled at the active edge of the clock edge. In other words, at every active edge, it is decided whether the counter will stay at the current state or will go to the next state or will go to the previous state.

Q: How do we initialize wires to certain values?

A: Wires can’t be initialized. Only way a wire can get input is, by testbench or by some other signal. Though, register/flip-flop can be initialized.

Q: In lab 6, I did not understand what was meant by 'same configurable logic'. Does it mean the same configuration file for both the projects?

A: It means the same design file. Note that the design file, contains logic tile, switchboxes, and their fixed connections. Different circuits could be configured using same design, by just changing the configuration file (memory content).

Q: *Just to confirm, do we have to use the same configurable logic used in lab5 in lab 6? I am asking this since in the document it is mentioned*
*"In lab 6, you implemented **three circuits** from the list using “the same” configurable logic."*

A: “The same” means - FPGA.v should be the same for all the three circuits of lab 6. It could be different from lab 5 design.