3C7 Digital Systems Design

Assignment 2

Aim:

The purpose of this session is to bring together elements from all your Labs, but particularly Lab F-G, and give you experience with developing and testing a larger sequential design and targeting it to the FPGA board. This assignment is worth 15% of 3C7.

Sequence Detector

You need to design a system that comprises an LFSR, a sequence detecting Finite State Machine (FSM), and a counter, in order to count the number of times a certain codeword (10, 12 bits long) is issued in the stream of bits generated by the LFSR in a full cycle of that LFSR. Each student will be given a different sequence/codeword, found in a separate handout `LFSR_code_setup.pdf`.

In lecture 6, we explored a Mealy model for a sequence detector which would detect the sequence '101' within a serial bit stream. Using this as a reference, write the Verilog description for your own sequence detector that uses Moore model.

Instructions:

1. You must re-use code from blocks previously implemented in 3C7 as much as possible, i.e. you **must** reuse the particular LFSR you were assigned in Lab-F, updating it for the bit-width and corresponding feedback you need for this assignment. For the counter, you may use your own tested module (Lab F/G) or one from the Chu book.
2. Include a ***functional diagram*** in your write-up clearly showing the hierarchy of your design and how modules are all connected. Wire names should reflect what is implemented in your design.
3. Design your FSM, using the lecture-7 slides to help you. Your write-up must include a state machine diagram, an explanation of how the state machine works, and an appropriate minimisation strategy.
4. Your coding style should follow a ***rising edge sensitive*** design.
5. Design a top level testbench that instantiates your top-level module. Design an appropriate set of test vectors to test your code detection system and its functionality.
6. In the write-up, define a test plan for your module – this includes how the individual modules are tested, design of appropriate test vectors and a validation plan (how you will check the correctness of the output). The test plan would then explain how the integrated module will be tested (and the above steps at the top level).
7. In your write-up, show a legible, well-structured waveform (i.e. relevant signals visible) that clearly demonstrate the correct operation of the system.
8. Target the working system to the FPGA board.
9. Exploit peripherals to demonstrate your working design. (e.g display the count on 7-seg display and LEDs, use buttons/switches to drive inputs)

10. Include a section in your report called “Demo”. Within this, use screen shots of your board to demonstrate that your design is working, how it operates etc. You may optionally include a separate video that demos your design on the board. You still need the section in the report though.
11. Clearly state how many times you found your given codework in the full 2^{N-1} cycles of your given LFSR. Clearly show that this number is somehow output or displayed on the board.
12. Explain in your write-up how you are controlling the clock and show where in the XDC file you are doing this.
13. Open the implemented design and view the Utilization Report. Take a screenshot of the "Register as Flip Flop" section and include it in your write-up.
14. Take a screenshot of the project hierarchy of your code.
15. Include the final bit file in your submission.

Submission:

1. **Full write-up** named Assign1_surnameinitial.pdf, e.g. called Assign2_shankers.pdf. (**Must be in pdf format**)
    a. **Labelled waveforms** in the write-up to demonstrate the design is working where appropriate
    b. **Include a screen grab** of your utilisation report, timing report, and other requested screenshots or information.
    c. **Include a section called “Demo”** which clearly outlines how to demo your design on the board (which switches etc to use for inputs etc). This will be verified against your submitted bitfile.

    NOTE: Use the **first link** (marked Assignment-2 writeup) to submit the pdf file.
2. **A zip file** named Assign2_surnameinitial_codes.zip, e.g., called Assign2_shankers_codes.zip containing the following
    a. **code for the modules and testbench** you have written such that a 3rd person can recreate your *bitfile*.
        i. **Note**: all code MUST be suitably commented, i.e. to a very high level.
    b. **Your final bit file** used for the Demo section of the report.

    NOTE: Submit the zip file using the **second link** (marked Assignment-2 code).

Deadline:

Both Groups: Friday 14th April 2023, 11 PM.
All submissions are via blackboard. Assignment-2 report is worth 15% of your CA.

Plagiarism

Plagiarism is interpreted by the University as the act of presenting the work of others as one’s own work, without acknowledgement. Plagiarism is considered as academically fraudulent, and an offence against University discipline. The University considers plagiarism to be a major offence, and subject to the disciplinary procedures of the University.

Visit http://tcd-ie.libguides.com/plagiarism

All students must complete the TCD Ready Steady Write plagiarism tutorial and sign a declaration when submitting course work, whether in hard or soft copy or via Blackboard, confirming that you understand what plagiarism is and have completed the tutorial. If you read the information on plagiarism, complete the tutorial and still have difficulty understanding what plagiarism is and how to avoid it, please seek advice from your College tutor, your Course Director, your supervisor, or from Student Learning Development.

e.g. having exactly the **same test vectors or module** as another person **is plagiarism**.

Plagiarism will result in loss of **ALL** marks for this assignment.