LAB 4 – Finite State Machines (FSMs)

Learning Objectives

- Use top-down design approach with sub-systems
- Implement basic FSM circuits on FPGA
- Use interacting FSMs and write effective testbenches for FSMs

There is a tutorial part for this lab which should be followed first before attempting the lab task. The tutorial shows VHDL examples for basic flip-flops and one VHDL style of describing finite state machines (FSMs) of Moore and Mealy types.

Design Task

You are required to design and implement a bit sequence detector and test on the FPGA board. Following specifications apply to the system.

- The system detects the presence of the 6-bit sequence **100101** (read from left to right) in an incoming serial stream of bits and counts how many times the sequence has been detected so far.
- The sequence detection must consider overlapping between two successive detections. That is, 00110010100101100 should count as two detections.
- The system must be driven by the internal 100MHz clock on the board (with appropriate clock division, if required) and **must employ a finite state machine (FSM) based approach** for implementation of the sequence detection. You are encouraged to examine different FSM design approaches first (single FSM vs. interacting FSMs) and select the best approach with some justification. **As a design constraint, a parallel detection approach (i.e., based on magnitude comparator on parallel input bits) must not be used and the sequence detection sub-system must "see the bits serially, one bit at a time".**
- The incoming serial bit stream is generated by the slides switches on the board and entered as chunks of 16 bits at a time. That is, the user sets the 16 slide switches to a particular input bit pattern and then presses a load push button (can be any of the buttons on the board) to enter the selected 16 bits to the system. The system should store this 16-bits internally (i.e., append to the previously entered 16 bits) and **process the bits serially** to perform the sequence detection and counting. User inputs data in chunks of 16-bits and sequence detection and counting should happen in real-time as the input bits are entered to the system. Note that the system must still detect if the sequence is split across two successive user inputs, as shown below:

  User input 1 (first chunk of 16 bits): 1001001010101100
  User input 2 (second chunk of 16 bits): 1011001110101010
  User input 3 (third chunk of 16 bits): 0101011010101010
  …
- The sequence detection output (i.e., how many times the sequence has been detected so far) must be displayed in real-time on seven segment display (SSD) on the board. You can assume that the maximum detection count is 99. Upon reaching 99, in the next successful detection, the output rolls over to 00.
- There must be a reset button to reset the system to initial state where the input bit stream is reset to all 0's and detection output becomes 00.
- State any additional assumptions you make in the report.

Submission and Assessment

You will be assessed based on a report submission followed by oral assessment. Submit a short report (electronically typeset) in PDF format via Blackboard containing the following. In case you don't achieve the full functionality, you can report evidence of partial functionality (i.e., correct operation of different sub-systems within your design) to receive partial credit.

- Short introduction to the task **[1 mark]**
- Block diagram(s)
  Must include the top-level and sub-systems up to the level of selected abstraction **[2 marks]**
- Explanation of the design strategy, any assumptions, and justification for the selected design strategy. **[2 marks]**
- Functional Simulation **[4 marks]**
  Testbench must generate serial test input pattern to show:
  - At least one non-overlapping detection
  - At least one overlapping detection
  - Detection count
  - Reset functionality
- On-board testing (photos in the report and demo during oral assessment) **[6 marks]**
  Must include the evidence of the following:
  - Non-overlapping detection
  - Overlapping detection
  - Reset functionality
  - Detection count on the SSD
  - Detection of sequence split across two chunks of 16-bit user input
  - Real-time detection operation
- VHDL descriptions **[2 marks]**
- RTL schematic **[1 mark]**
- FPGA resources **[1 mark]**
- Conclusions **[1 mark]**
- Oral assessment **[4 marks]**

The report should not be just a collection of screenshots, but with proper explanation of figures/results with some discussion. The report is **due on Monday 16/09/2024 4:00 pm AEST**.

Oral assessment: Conducted during scheduled lab sessions in week 9. You need to explain your design to teaching staff and ***demonstrate on-board testing of your design*** followed by some Q&A to check the understanding of your work. Marks for the oral assessment will be distributed as follows:

0: No knowledge of the design **(total mark capped at 0)**
1: Very little knowledge of the design **(total mark capped at 50%)**
2: Reasonable knowledge of the design
3: Good knowledge of the design
4: Excellent knowledge of the design

Total mark for this lab is out of 24, subject to oral assessment caps as above and any late penalties as outlined in the ECP. If you do not attend/complete the oral assessment, your marks for this lab will be capped at 0, as indicated above.