Lab 4: Design and Implementation of FSM

Exercise 4.1: Instructor will explain about implementation of FSM in Verilog. Proceed further only after the explanation.

Exercise 4.2: Problem Statement: Design a state machine to design an UP-Down 4-bit even counter. The state machine has one input **(CountUP)** and one 4-bit output (**CountValue**). When **(CountUP=1)** the counter will have sequence 0-2-4-6-0.. If **(CountUp=0)** counter will have sequence 6-4-2-0-6... In addition to **CountUP** there are two more inputs **Clk** and **Reset** for normal operation of FSM. When **Reset** is enabled **CountValue** is reset to 000**.** The counting rate is (i.e. the time between two successive states is 0.5s).

(Please refer to the file named "**Vivado_Design_Flow_All_Steps.pdf**" for a review of all the steps in the design flow)

1. Create a Vivado Project and write Verilog code (**CountFSM.v**) for implementing the above FSM. (write this verilog code in your observation book)

2. Write the Test Bench **Count_Test_FSM.v** and simulate your design to check the functionality.

3. Add clock division code to **CountFSM.v** such that the actual input **Clk (Y9 pin with frequency of 100MHz)** is converted to Clock of frequency 2Hz. This 2Hz signal is used as clock for running the FSM. Connect the new (2Hz) clock as one of the outputs of **CountFSM.v** for reference.

4. Plan your I/O mapping (using **I/O planning** option) such that actual input **Clk** is connected to internal clock pin **Y9**, **Reset** is connected to push button switch, other input (**CountUP**) is connected to DIP switches and outputs are connected to LEDs. In the ZedBoard, the pin numbers indicating the DIP switches, LEDs and internal clock are listed in table below. Save the mapping information as **CountFSM.xdc**.

5. Synthesize (**Run Synthesis**) and Implement the design (**Run Implementation)**.

6. **Generate Bitstream** and port your design on to FPGA (**Open Hardware Manager→ New Target→… Program Device**)

7. **Check the output on FPGA.**