# UNIVERSITY OF TEHRAN
## Electrical and Computer Engineering Department
## Digital Logic Design, ECE 367 / Digital Systems I, ECE 894
## Spring 1400-01
## Computer Assignment 5

Synthesis of Counters, Shifters and State Machines - Week 13

| | |
|---|---|
| **Name:** | **Date:** |
| **Username:** | |

A serial transmitter circuit searches on its *serIn* input for a start-sequence to begin transmitting its *serIn* on its *serOut*. When the start-sequence is received, the *serOutValid* is asserted and for the next 1024 clock cycles whatever appears on *serIn* will be transmitted on *serOut*. After the entire 1024 bits are transmitted, the circuit returns to the state where search for the start-sequence begins again. The initial state of this circuit is where it searches for the start-sequence. This machine has two parts. One is the start-sequence detector, and the other is the input to output transmitter.

The start-sequence consists of 00, followed by nine 1's and then another two zeros (0011111111100). When the start6-sequence ts detected, the control goes to the transmitter part and when the transmitter completed its task, the start-sequence detector returns to its first state searching for another start-sequence on *serIn*.

The transmitter waits for a signal from the start-sequence detector. When received, it puts the next 1024 seriasl data on serIn to serOut. When completed, it issues a signal to the start-sequence detector for it to start searching for the start-sequence again.

a. For implementing the start-sequence detector, you need a Moore machine and a divide by 9 circuit. Write SystemVerilog description of this part using a sequence tetecor and a divider. Both parts are to be written in SystemVerilog and the two should be completed as a unit to be used in the next part along with the transmitter part.
    i. Using a SystemVerilog testbench in the ModelSim simulation environment completely simulate your circuit. This is your pre-synthesis description.
    ii. Import your start-sequence detector in Quartus and build a symbol for it. Synthesize this design and see its timing, floor-plan and cells used. The synthesis process generates the post-synthesis description of your start-sequence detector. The .vo and .sdo files that are produced contain the netlist and timing of the synthesized circuit.
    iii. Instantiate the pre- and post-synthesis descriptions of the start-sequence detector in a SystemVerilog testbench and compare the timing of the two descriptions.

b. In Quartus, use existing library components, i.e., *lpm*, and other necessary gates and structures to design the transmitter part of your design. Use an asynchronous reset and

the positive edge of the clock. This is your pre-synthesis design of the transmitter part of your circuit.

i. Build a symbol for the transmitter circuit. Synthesize this design and see its timing, floor-plan and cells used. The synthesis process generates the post-synthesis description of the transmitter circuit. The .vo and .sdo files that are produced contain the netlist and timing of the synthesized circuit.
ii. Instantiate the post-synthesis transmitter circuit in a SystemVerilog testbench and observe its timing.

c. In a new Quartus project, import the descriptions of Part a and Part b to build the serial transmitter circuit described in the problem description.

i. Synthesize this design and see its timing, floor-plan and cells used. The synthesis process generates the post-synthesis description of the serial transmitter circuit. The .vo and .sdo files that are produced contain the netlist and timing of the synthesized circuit.
ii. Instantiate the post-synthesis serial transmitter circuit in a SystemVerilog testbench and observe its timing and its functionality.

---

**Deliverables:**

Generate a report that includes all the items below:

A. Prior coming to the lab, for all the above problems hand-drawn schematic diagrams and partial timing diagrams are required. Your SystemVerilog descriptions must correspond to the circuit diagrams. Your simulation run and the project built for this purpose must be demonstrated to the TA.
B. Document your Quartus projects of Parts a, b, and c. Make sure you understand the synthesis outputs and their corresponding timings.
C. For all three parts, you should look at the FPGA layouts, device view and RTL view. Be able to explain the details of various views. In the layout be able to identify FPGA cells that use a memory element versus those that are purely combinational.
D. For all problems, be prepared to answer questions asked about the timings, generated hardware, pre- and post- synthesis, and FPGA mappings.

Make a PDF file of your report and name it with the format shown below:

*FirstinitialLastnameStudentnumber-CAnn-ECEmmm*

Where *nn* is a two-digit number for the Computer Assignment, *mmm* is the three-digit course number under which you are registered, and hopefully you know the rest. For the *Firstinitial* use only one character. For *Lastname* and for the multi-part last names use the part you are most identified with. Use the last five digits of your student id (exclude 8101) for the *Studentnumber* field of the report file name.