UNIVERSITY OF TEHRAN

Electrical and Computer Engineering Department

Digital Logic Design ECE 367 / Digital System I ECE 894

Fall 1399

Computer Assignment 5

Counters, Shifters, State machines - Week 13

Name:	**Date:**
Username:	

In this assignment you will design a Multi-channel Synchronous Serial communication Demultiplexer MSSD. The source provides a one-bit stream of data that includes the size of data to be transferred, the destination port, and the actual data.

Problem Description: Serial bits of data appear on the serIn input of MSSD. Normally, in no-transmission mode, *serIn* is 1. Transmission begins when serIn makes a 1 to 0 transition. The 0 value bit marks the beginning of transmission and has no other information. The two bits that follow are the port number, *p*, LSB first in time. The next 6 bits are the number of bytes, *n*, LSB comes first in time. With $1 + 2 + 6 + n \times 8$ clock cycles after *serIn* becomes 0, it will return to 1 and another transmission begins with another start-bit. Data on *serIn* are synchronized with the MSSD clock, *clk*.

This multiplexer extracts the destination code (*d*, two bits) and the number of bytes that will go to the destination port (*n*, 6 bits). After that, the next $n \times 8$ bits will be transmitted to port *d*. In addition to the four ports of MSSD, i.e., *p0*, *p1*, *p2*, and *p3*, the active port id, i.e., *d*, and an *outValid* signal are the outputs of MSSD. The *outValid* signal remains 1 for as long as actual data is being transmitted to an output port.

1. Show block diagram of MSSD using state machines, counters and shift-registers.
2. Describe all components of your design in SystemVerilog and simulate the complete design to verify its operation. Use a testbench in ModelSim. This phase of simulation is referred to as pre-synthesis simulation.
3. Take your design into Quartus and implement it using separate modules as you did in Part 1. For each module you have the choice of using your own SystemVerilog description or using Quartus modules, e.g., *lpm* modules. It is recommended that you describe the main controller of your system in SystemVerilog using Huffman modeling style.
4. Synthesize your design in Quartus and obtain .vo and .sdo files. These files are your post-synthesis netlist and timing files.
5. See synthesis reports, view various diagrams that the synthesis tool generates.
6. Add the post-synthesis output of Part 4 to the testbench of Part 2 and simulate these two implementations of MSSD alongside each other.
7. Provide simulation reports and compare the timings of the two implementations.