UNIVERSITY OF TEHRAN
Electrical and Computer Engineering Department
Digital Logic Design, ECE 367, Fall 1398
Computer Assignment 5
RTL Datapath and Controller
Week 13

Name:
Date:

In this assignment, you will be implementing the complete SMBS of Computer Assignment 4 with the exception that the number of bits in a chunk of data will be transmitted over the serial input instead of a fixed value (i.e., 5 in CA 4). The problem description follows.

The incoming data is serial on *serIn* that is synchronous with the input clock, *clk*. When there is no incoming data, *serIn* stays at 1. After this, the first 1 to 0 transition indicates serial data is starting. In this case, the first 0 is just used as the start bit. Following the start bit (0), the next six bits indicate the destination port(s) (i.e., first four bits) and the line number to broadcast data on (i.e., the next two bits). In both cases, the least-significant bits come first in time.

Immediately after specification of the destination, another 6-bit number (LSB first) appears on *serIn* that indicates the number of bits in a data chunk (a chuck can contain from 0 to 64 bits). We refer to this number as c. A data chunk of size c, is a group of c consecutive synchronous data bits that are followed by a spacer. The spacer is one clock duration 0 on *serIn* that is synchronized with the system clock.

Immediately after transmission of c, chunks of c-bit data followed by a 0 spacer begin. The data received will be sent to the specified port. The spacer will not be sent to the destination. This continues for as long as a 0 spacer is received after c bits of actual data. If a 1 is received instead of a 0 spacer, this indicates a stop-bit and transmission to the specified port ends. For as long as *serIn* remains at 1, no data is received. Once again, the first 1 to 0 transition is an indication of a 0 start-bit, after which bits for the next destination port start arriving.

Your system has a *valid* output that becomes 1 when valid data is being transmitted on one of the output ports. This signal is 0 during the idle time, start bit, stop bit and the spacer. Another output of the system is *outPort* that is a 6-bit number indicating which output data is being transmitted on.

1. Write the behavioral description of the transmitter logic of CA 3. This should be done in Verilog (using **always** statements is recommended). After testing it in ModelSim bring it into Quartus and use it as a component.
2. Design the complete datapath of your SMBC design using the above component and RTL components discussed in class.
3. Design the controller of SMBC using counters and state machines.

Implementation Phase:

A) Build the datapath of your design in Quartus using predefined Intel Altera components, Verilog modules, or discrete parts.
B) Describe the circuit's controller in Verilog.
C) Generate a symbol for the datapath from its schematic implementation.
D) Generate a symbol for the controller from its Verilog and/or schematic description.
E) Generate the complete SMBC design by instantiating its datapath and controller.
F) Synthesize SMBC circuit and generate its .vo and .sdo files.
G) In a testbench, test your complete circuit.