Homework 5

Please turn in all verilog code and a hardcopy of your simulation results. You can demonstrate proper operation by either a printout of the "monitor" output (the table printed once the simulation has been run) or by a printout of the waveforms (the results of some problems are easier to see one way and the results of other problems are easier to see another). **Be sure to annotate your simulation results telling me how your results prove that you have met all specifications.**

The Problem:

Envision a stream of 4-bit data packets that is traveling across an asynchronous channel. The source of the information has two 1-bit communication ports, **READY** and **DONE**. This configuration is illustrated in figure 1:

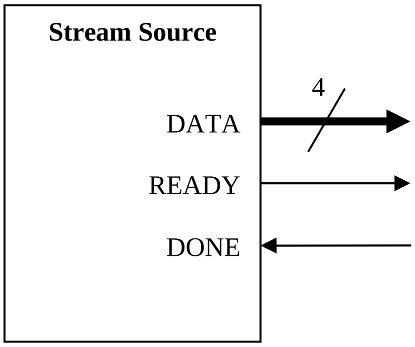

Figure 1: Interface of Stream Source Module.

When the Stream Source Module (SSM) has a new packet of data available, it asserts the active-high **READY** port. It *continues* to assert **READY** until the **DONE** port is asserted, telling the SSM that the data has been received and that it is ok to produce a new packet of data.

Your job is design, implement, and test a Stream Receiving Buffer Module (SRBM) that

- Communicates with the SSM via **READY** and **DONE**
- Stores up to two packets of data
- Outputs a stored packet of data on an output line **OUT**
- Communicates with a third device through the ports **NEW** and **UPDATE**

The SRBM asserts **DONE** when it has successfully stored the new data packet from the SSM. It outputs the next packet in sequential order of arrival on the 4-bit output **OUT**, asserting **NEW** whenever new data is available. When the third, unknown device has collected the data from **OUT**, it will assert **UPDATE**, signaling to the SRBM that it can output the next packet of data.

Note that at any time the SRBM could be in one of the following situations:

- Both packets are stored, waiting for the third device to signal **UPDATE**
- Both packets are empty, waiting for the SSM to signal **READY**
- One packet stored, transmitting to third device while storing a second packet from the SSM
- etc.

1. Draw a top-level interface diagram of the SRBM
2. Divide your SRBM into two sub-units: a data partition and a controller partition
3. Outline the responsibilities of each of your two sub-units
4. Draw interface diagrams for each of your sub-units
5. Draw lower-middle level schematics of your sub-units. If you plan to use a finite state machine, include a block called FSM in your design – you don't have to go deeper into your schematic.

YOU WILL NOT RECEIVE CREDIT FOR ANY OF THE FOLLOWING PARTS IF YOU HAVE NOT FINISHED THE ABOVE PARTS

6. Implement your sub-units in Verilog. Use multi-module design whenever convenient.
7. Write a test-bench for your Verilog.
8. Simulate and annotate your work.