EE 533 Lab #8
Convertible First-In-First-Out (SRAM) Memory
Young Cho - youngcho@isi.edu

You must design and integrate a special FIFO to allow data to flow between the NetFPGA reference pipeline designs and your processor. The specifics of the FIFO, such as the width of data I/O and control interface, should be dependent on your processor and how you have chosen to interface with external components. However, it is expected that the FIFO should be designed with Block RAM modules that are instantiated as dual-port SRAM modules, like how Drop FIFO was designed in the Mini-IDS laboratory.

One of the address and data ports can be designed as the input interface to the FIFO. The other port can be used as the output. As with the Drop FIFO, the addresses of the head and tail of the FIFO must be tracked using two accumulators, registers, and control logic. Although your FIFO does not have to perform the drop function, it is expected to identify the head and the tail of each packet, then (1) buffer all of the data and the control signal for one network packet, (2) when one packet is buffered in the FIFO, send a FIFO FULL signal to the circuit that is connected to the input so that next packet data is stalled, (3) send some sort of signal to the processor that the data is ready to be processed.

The processor should have read and write access to head and tail address registers and full access to the SRAM that has the data. One way to give access to these components is by instantiating Multiplexors at the inputs and selecting between the NetFPGA datapath and your processor's MEM stage of the pipeline. You can modify the design so your processor can interface with the FIFO. The following is the block diagram of the described FIFO.

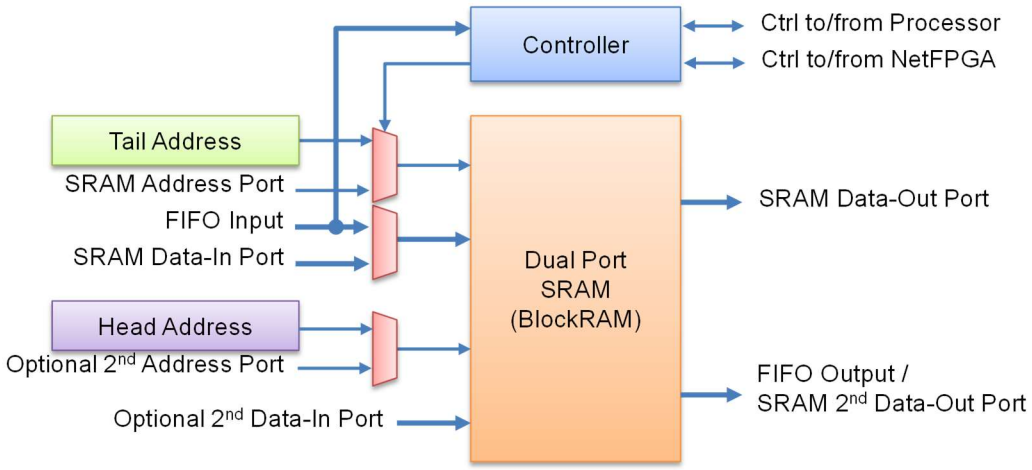

Figure 1: Convertible FIFO/Dual Port SRAM

Once the design is simulated, you must integrate and test the FIFO with the processor in NetFPGA in DETER. As part of the Lab, you must write and execute a few small codes to test your single-core Network Processor. Your codes should test reading and modifying the content of the packets. You should also test the limits of your network processor, such as the maximum throughput of the code, etc.

Submit the source codes, simulation output, test code, and description of your test setup and the results. You must also demonstrate test code running on your network processor.