Derek Tzeng

18-341

Project 2: Network on Chip

Design of Node

In my node, I have 3 threads:

1. Deals with the loading of the queue/5th register from the testbench
    - If there is a packet available, this thread checks if the queue and 5th register are both empty. If they are both empty, then the packet will go directly to the 5th register and "output_buffer_full" will be asserted. Otherwise, it will be added to the queue.
    - If there is no packet available, then this thread will read a packet into the 5th register if the queue is not empty, and assert "output_buffer_full".
2. Send the contents of the 5th register to the router
    - This thread waits until "output_buffer_full" and "free_outbound" is asserted, then writes the contents of the 5th register byte by byte to "payload_outbound".
3. Receive packets from router and send them to the testbench
    - This thread waits until "put_inbound" is asserted, then reads the 4 bytes one after one, and send the whole package right after.

The diagrams for the first 2 threads are on the next page, and the last thread is on the following page.

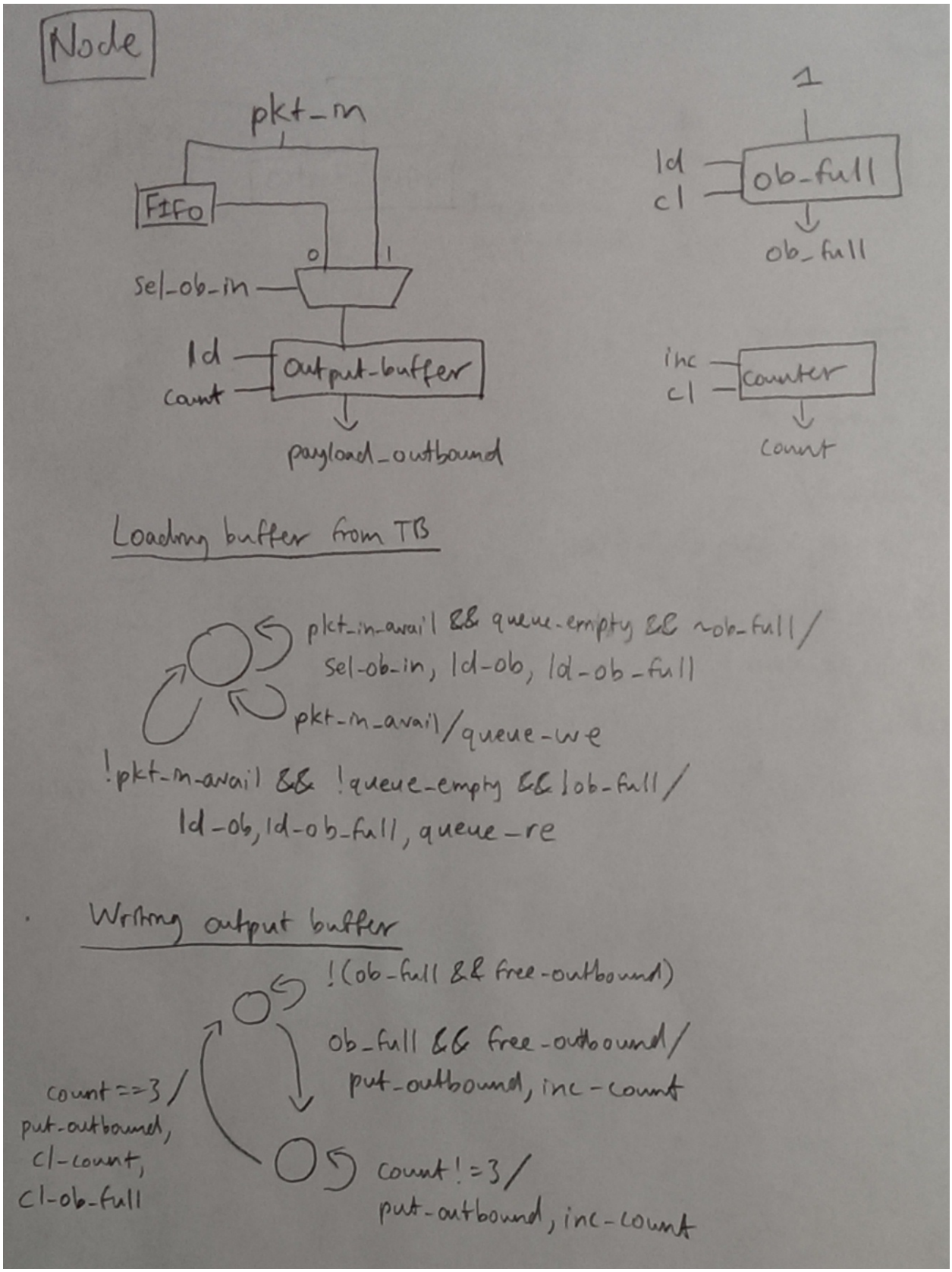

Figure 1 – Node threads 1 and 2

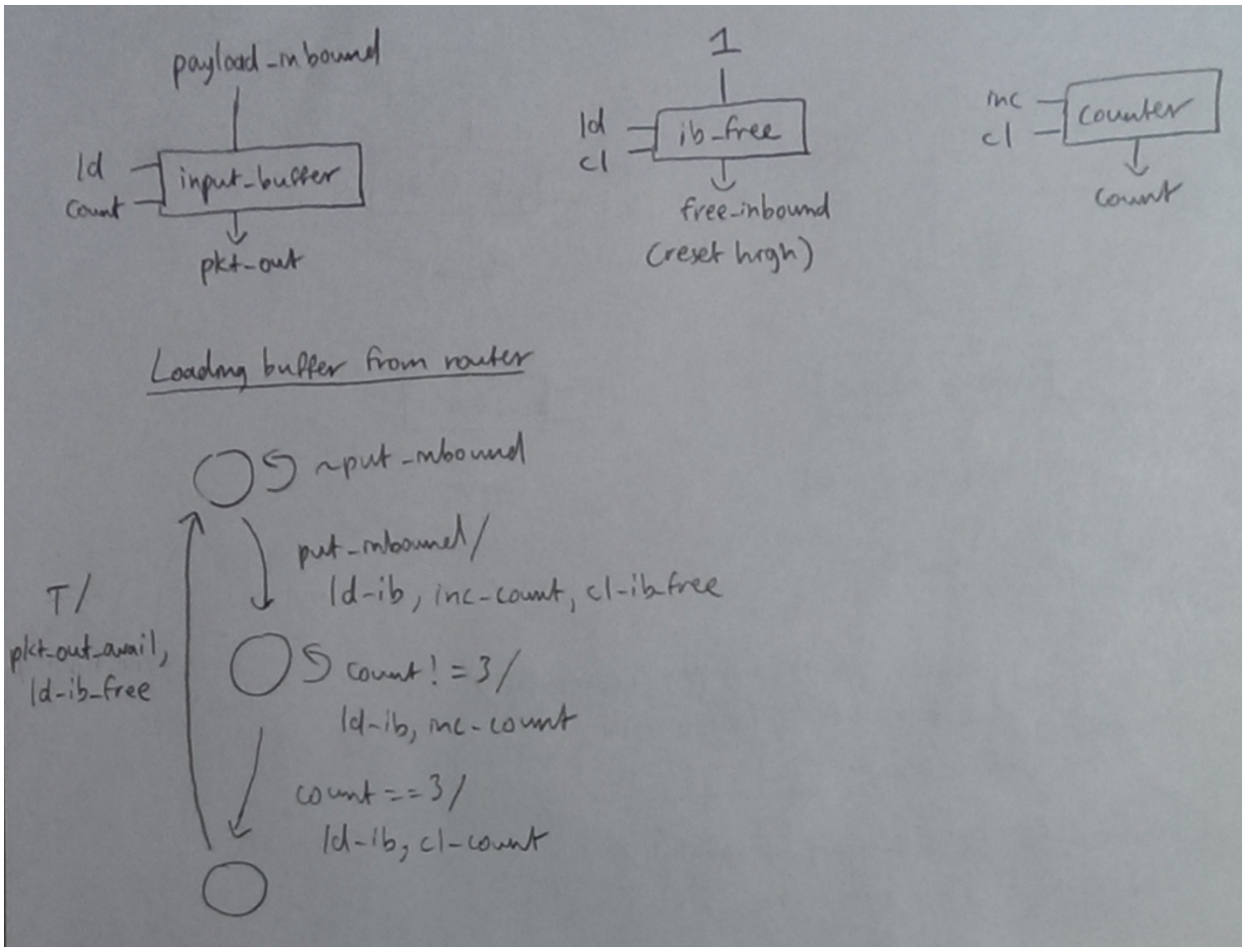

Figure 2 – Node thread 3

Design of Router

To design my router, I first had a pair of threads for each port that dealt with reading into an input buffer and writing from an output buffer.

The reading thread waits for a “put_inbound” on the corresponding port and loads the input buffer while setting the “free” and “valid” status points accordingly.

The writing thread waits for the output buffer to be full and “free_outbound” on the port and writes to the outbound while clearing the “full” status point accordingly.

The diagram for these threads is on the next page.

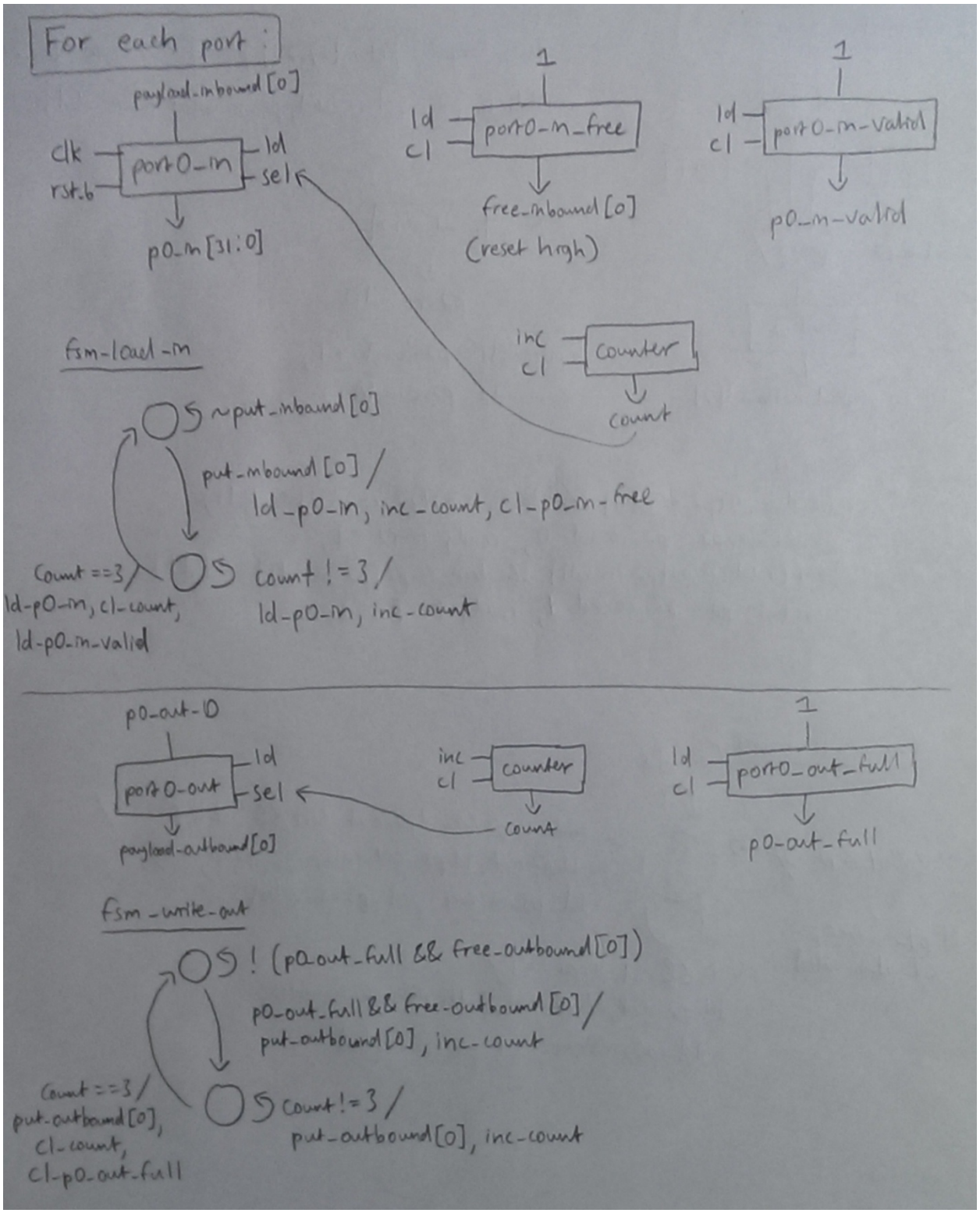

Figure 3 - Router reading/writing threads

After the reading and writing threads, I can now design the routing logic.

For each output buffer, I have a mux that feeds the input buffers of the other 3 ports into this output buffer in clockwise order. I also keep two arrays: one that contains all the destinations of the 3 input buffers, and another one that contains whether or not the 3 input buffers are valid. These arrays are also in clockwise order starting from the output buffer port. This way, I can have a counter that keeps track of where the first priority starts in a round robin fashion. Every time something is ready to be loaded into the output buffer, I will increment the counter so the priority goes to the next port. When I load into the output buffer, I will also update the appropriate status points of the source port according to the counter.

The diagram for the routing logic is on the next page.

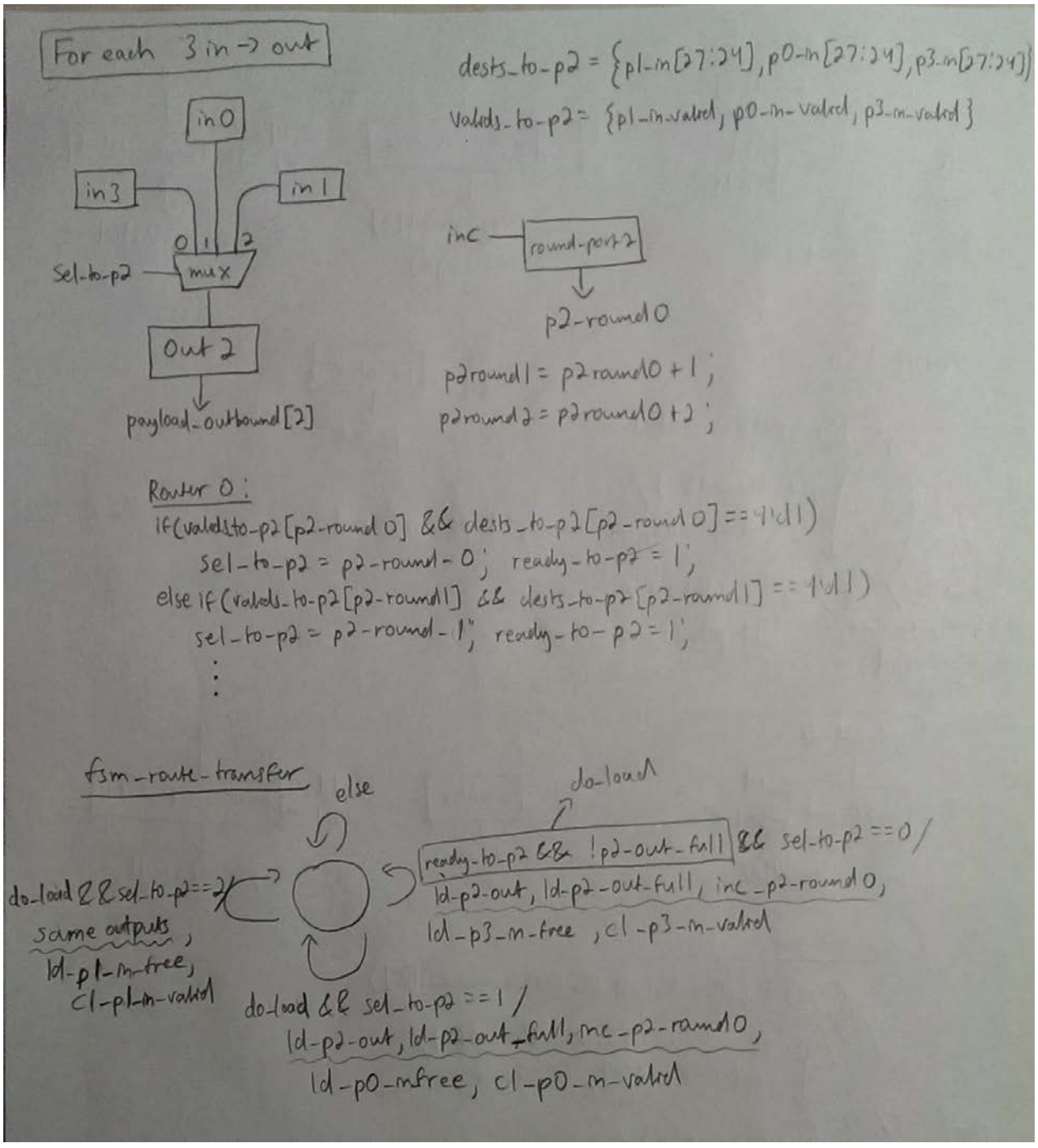

Figure 4 - Router routing logic