**Chapter 7, Threads and Interprocess Communication homework 2**

Modify the following design to incorporate a 2nd consumer that is provided integers by the same producer via a mailbox. Assume the 2nd consumer has the same latency as the 1st consumer. Similar to the Consumer class below, print out a message before/after the consumer calls get.

`default\_nettype none

module top;

test test();

endmodule

|  |  |
| --- | --- |
| `default\_nettype none  program automatic test;  mailbox #(int) mbx;  class Producer;  task run();  for (int i=1; i<=3; i++) begin  $display("%0t: Producer: before put(%0d)", $time, i);  mbx.put(i);  $display("%0t: Producer: after put(%0d)", $time, i);  end  endtask  endclass // Producer | Consumer c;  Producer p;  initial begin  // Construct mailbox, producer, consumer  mbx = new(1); // Bounded mailbox - limit 1!  p = new();  c = new();  // Run the producer and consumer in parallel  fork  p.run();  c.run();  join  end // initial  endprogram |
| class Consumer;  task run();  int i;  repeat (3) begin  mbx.peek(i); // Peek integer from mbx  #10ns;  $display("%0t: Consumer: before get(%0d)", $time, i);  mbx.get(i); // Remove from mbx  $display("%0t: Consumer: after get(%0d)", $time, i);  end  endtask  endclass // Consumer | |

Deliverables:

1. Code for the package, program, top level module, the golden model, and the interface.
2. Waveform of the directed test and the next 10 random cycles (this will take multiple pages) of the following signals in the golden model you created.
   1. error count
   2. req, grant, and enable for each port
   3. priority

Deliverables:

1. Code for the program.
2. Copy of transcript window.