Network On Chip

Vishavjit Parihar, Dr. Mostafa Abdelraheim

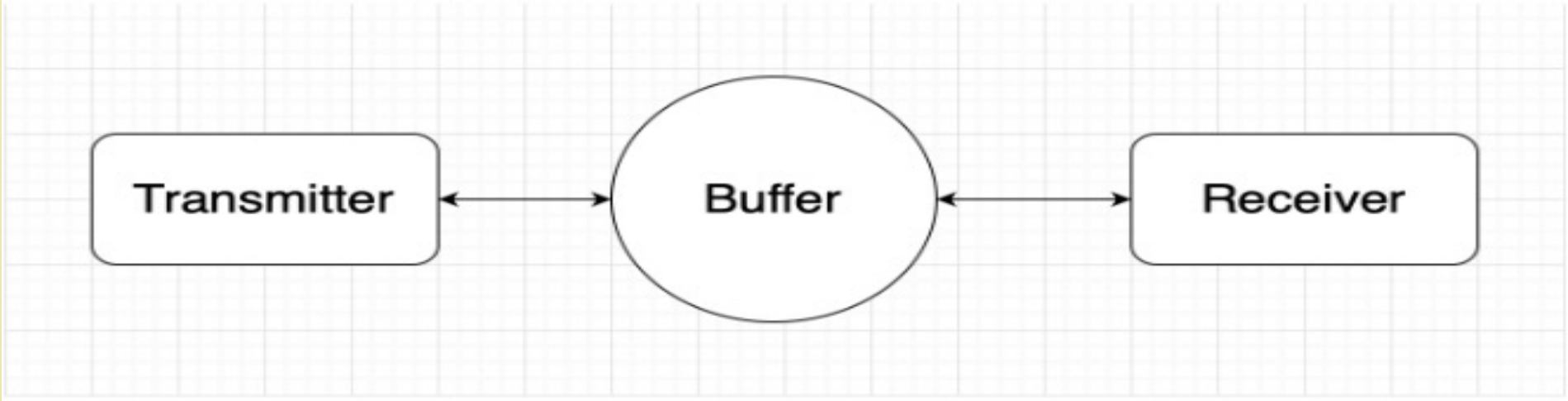
Introduction

A network-on-chip(NoC) is a network-based communications subsystem on an integrated circuit, most typically between modules in a system on a chip(SoC). The network on chip is a router-based packet switching network between SoC modules. NoC technology are way more reliable and faster than that of the traditional BUS or Crossbar Communication architectures. NoC will also improve the Scalability of SoC, and power efficiency of the SoC as compared to the other communication subsystem design.

Objective

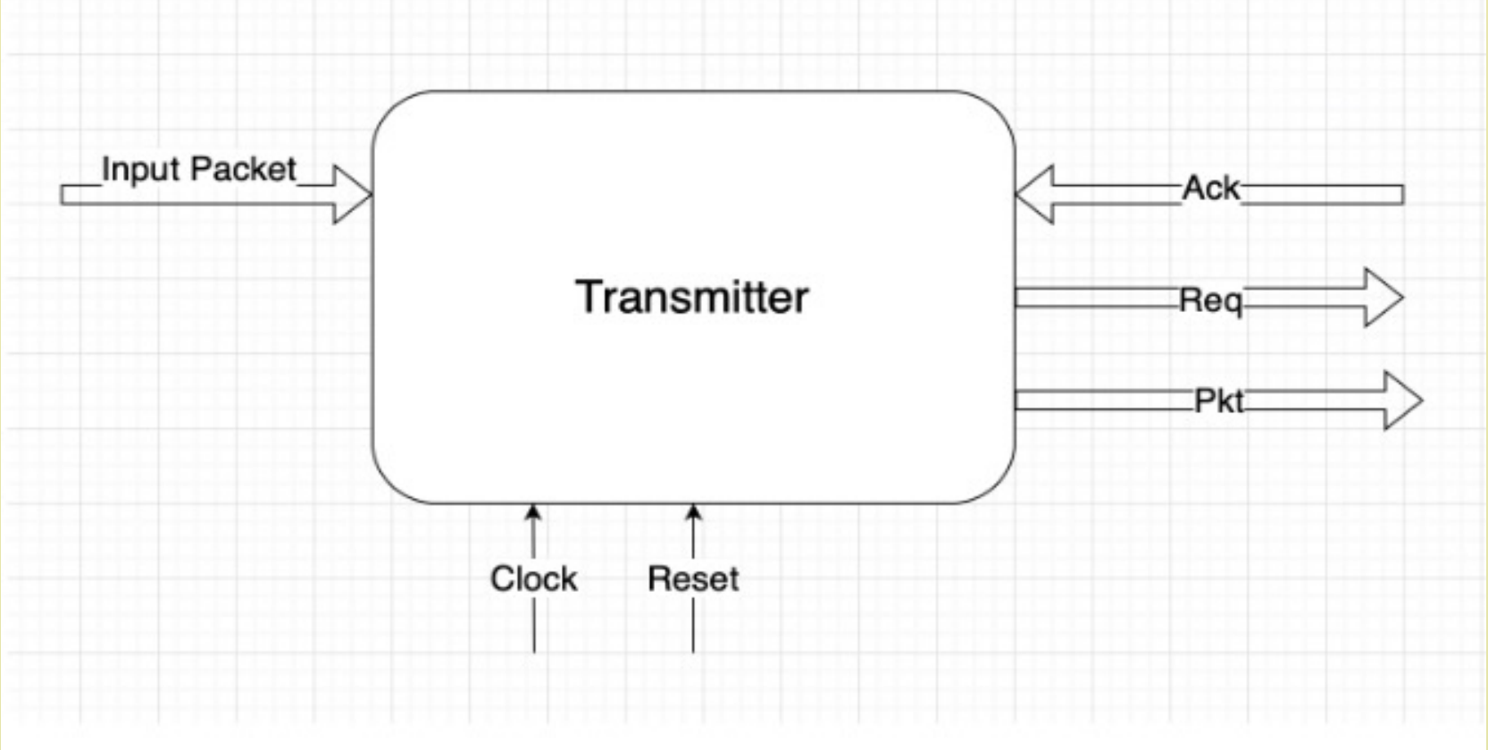
- To develop a module that can take inputs from a user, store that given input if output terminal is not ready to send that output back to the user.
- To develop a fast and safe means to transferring data in a Network of Chips

Method



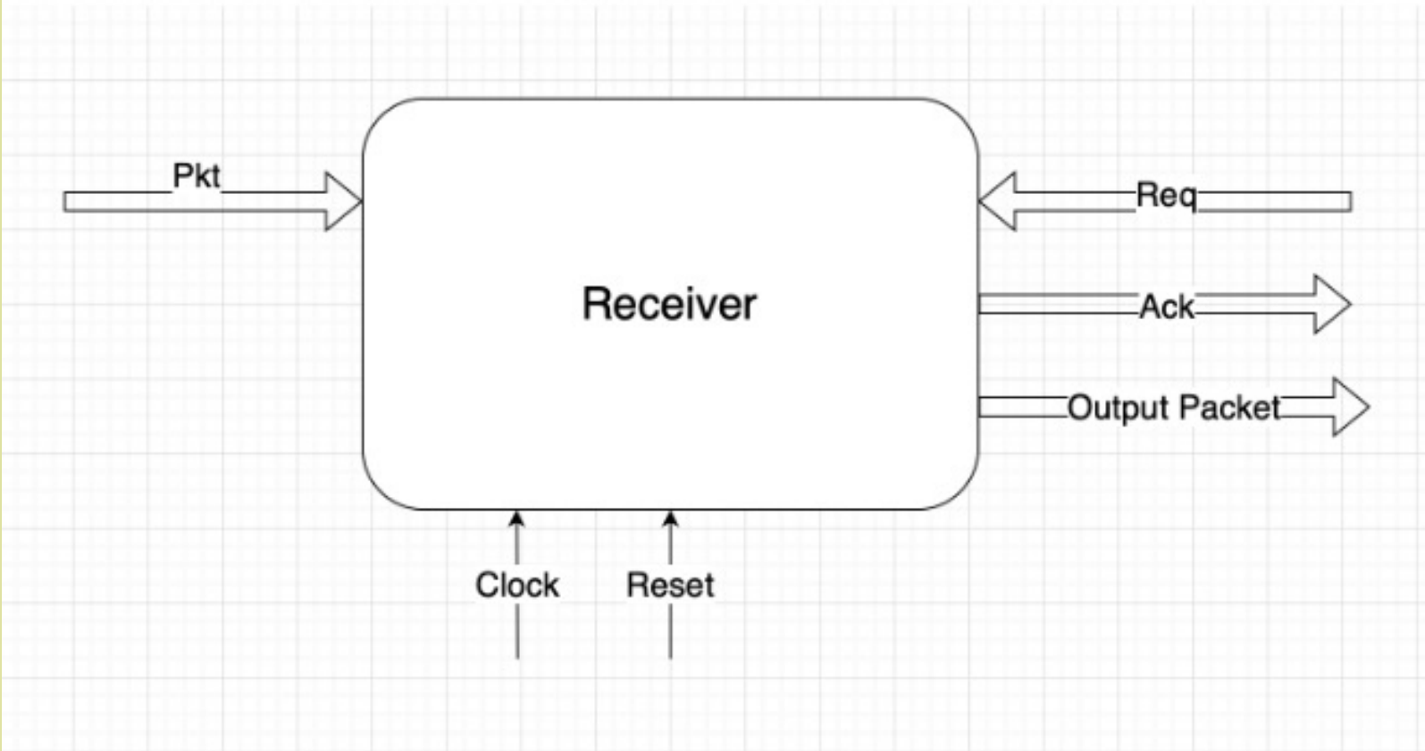
- Designing the Module in VHDL
- Implement Handshaking Process for the inner modules to communicate

Transmitter



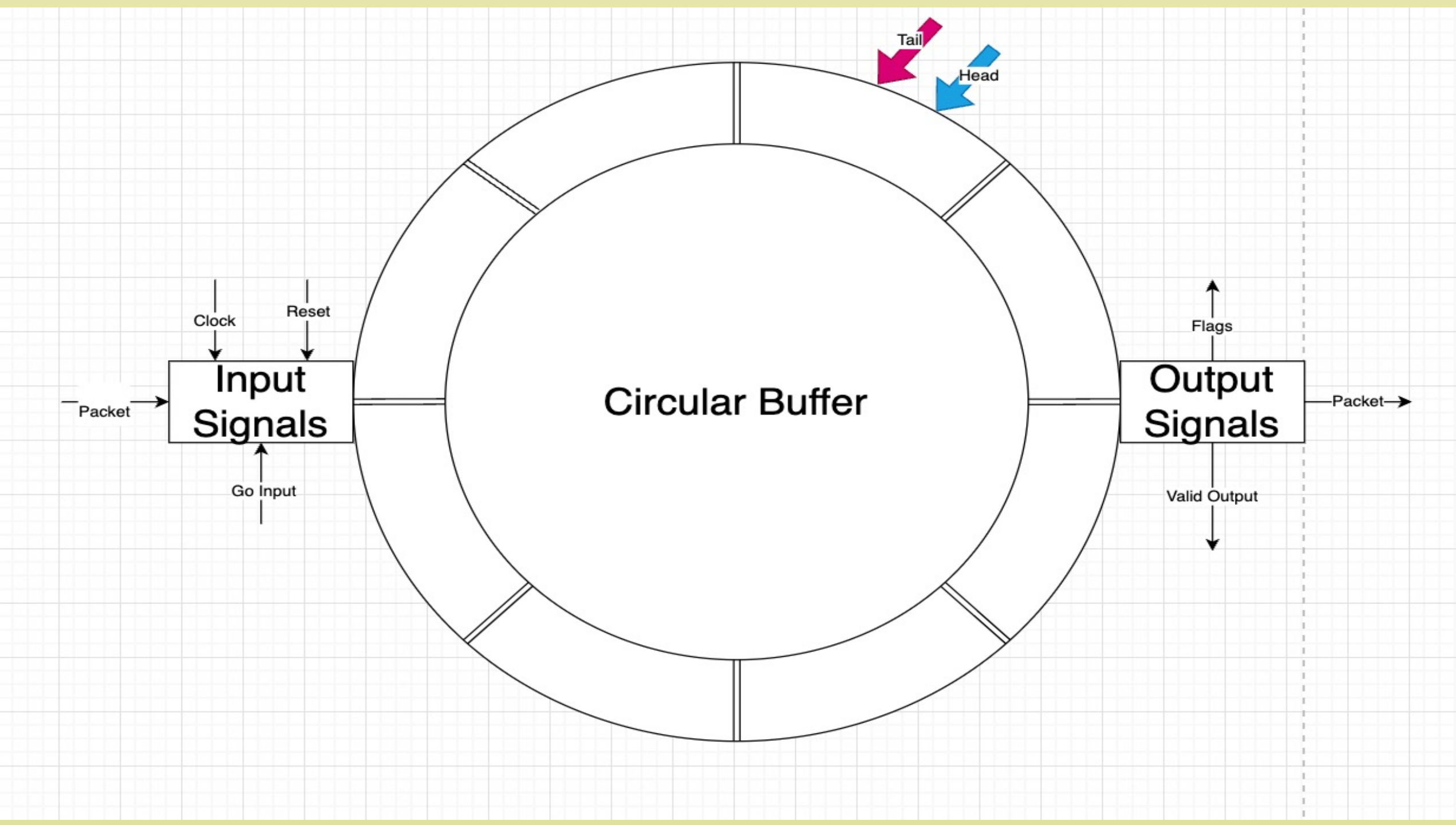
Transmitter has 4 inputs and 2 outputs. Transmitter works like a state machine, states will change depending on Input port and Ack port. Clock and Reset are there just to keep this component in sync with the other component of the Module. Transmitter will receive packet from user and send the packet to the buffer.

Transmitter



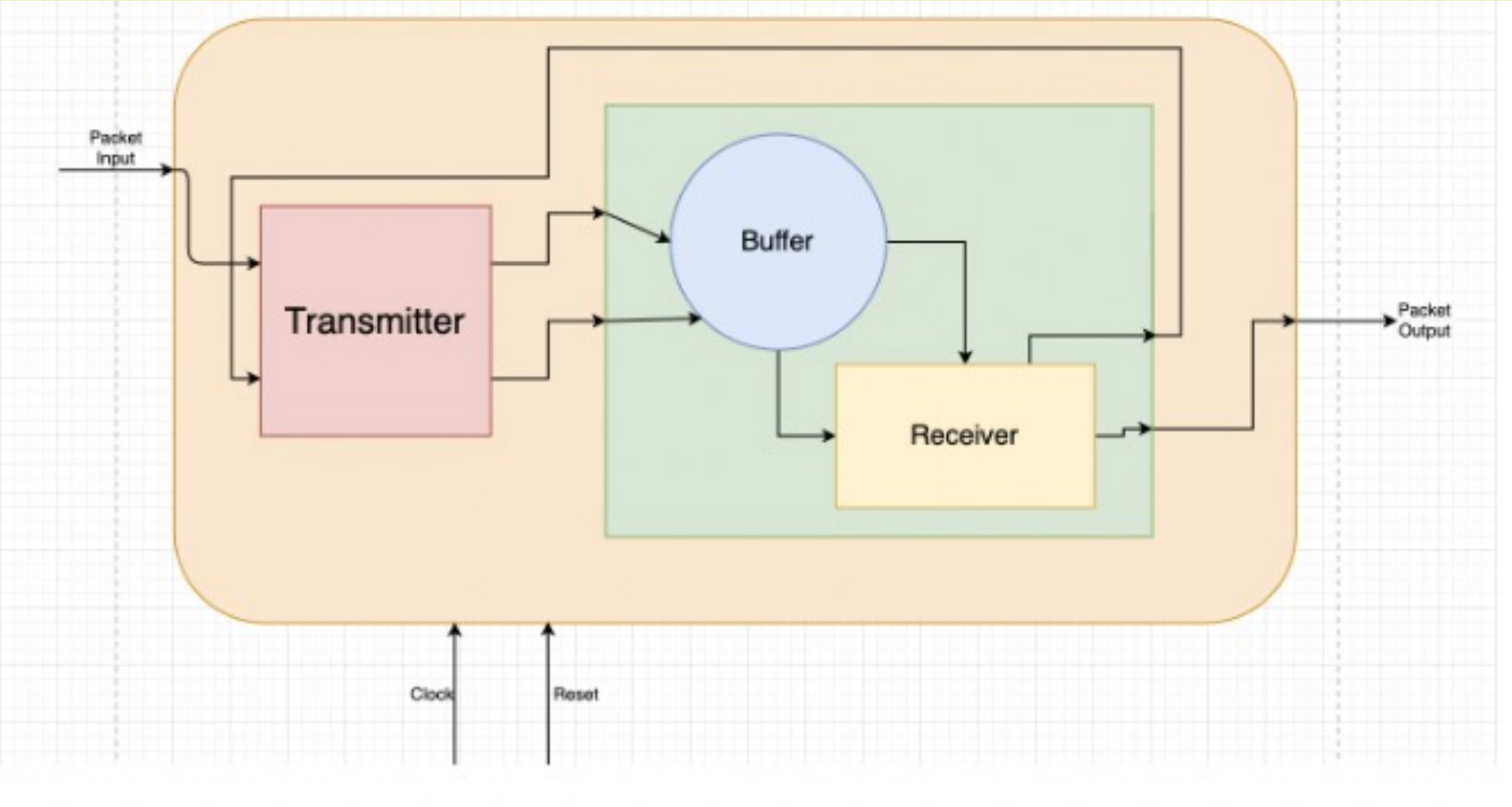
Receiver has 4 inputs and 2 outputs. Receiver works like a state machine, states will change depending on Pkt port and Req port. Clock and Reset are there just to keep this component in sync with the other component of the Module. Receiver will receive packet from Buffer and send the output to the User.

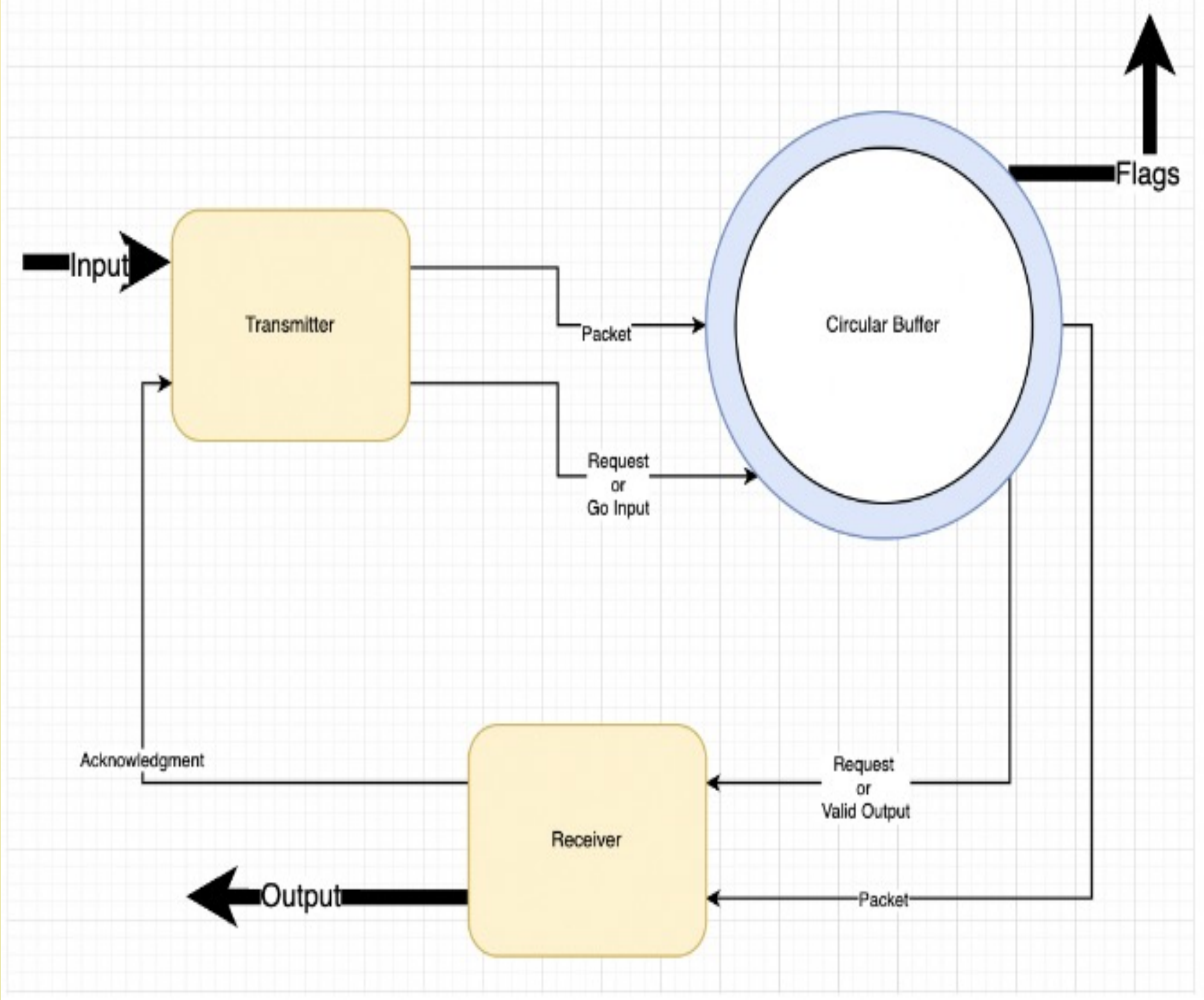
Circular Buffer



Circular Buffer is a FIFO type of Buffer. This Circular Buffer will accept input packets and it will keep those packets in order. Thus, the first packet into the buffer will be the first packet to go out of the buffer. This buffer will have 4 Inputs and 3 Outputs. The packet input of buffer will come directly from the Transmitter's Pkt. The Go Input input of the buffer will come directly from the Transmitter's Req. Clock and Reset are there just to keep this component in sync with the other component of the Module. In the diagram our Circular Buffer has only 8 slots to save data. There is also Tail and Head component of the buffer. Head and Tail are needed to keep track in which slot of the buffer the packet is stored. Moreover, Head and Tail's position will determine if the buffer is empty or full. The Flag output signal will just tell the user when the Buffer is empty or full. The packet output signal will carry the output signal to the Receiver. The Valid Output signal will also go to the Receiver as the Req signal.

The Working Module

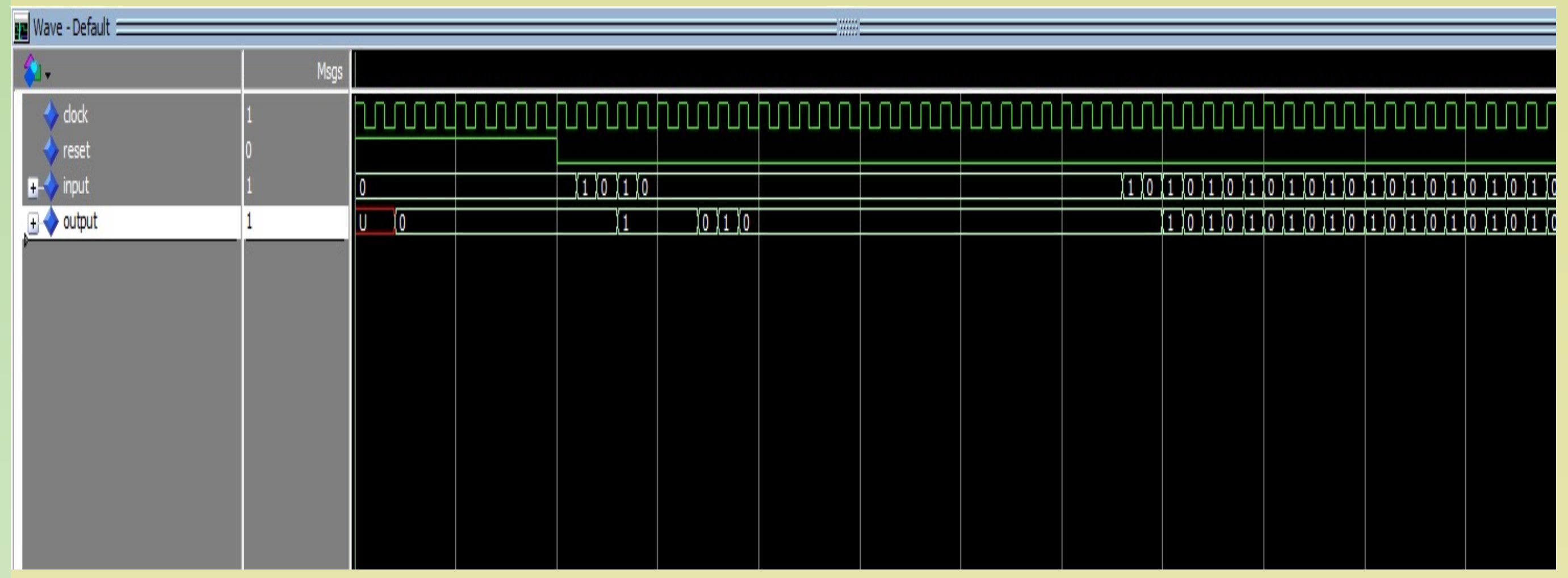




All of the components of module are connected in the given manner such that transmitter is connected to circular buffer, and circular buffer itself is connected to the receiver, and receiver is return is connected to the transmitter itself. In general, our whole module has 3 inputs and 2 outputs.
The 3 inputs are – Input Packet, Clock, and Reset
The 2 outputs are – Flags, which will include empty flag, almost empty flag, full flag, and almost full flag, and the other output is the Output Packet. Everything else is internal signal inside the module.

Once the user sends the input to the transmitter, transmitter will accept the input, and change its state while setting up the request signal on for the buffer. Transmitter will send the packet to the Buffer, not only that the request signal will also be sent to the buffer. This request signal will tell buffer that there is an input coming. Buffer will store the input and move the head signal to the next slot. In the mean while Transmitter will get another packet and same process will continue until the head signal has filled out all the slots. Which will mean that buffer is full, at that time buffer will start sending out packets to the receiver using the tail signal. With every packet sent – buffer will send a conformation to the receiver stating that the output packet was sent successfully. Receiver will receive the packet and the conformation signal. Receiver will send the packet out to the user and send an acknowledgement signal back to the transmitter – which will complete the cycle.

Results and Simulation



	Msgs
clock	1
reset	0
input	1
output	1

As we can see in the waveform, when the input packet is sent, after a couple of cycles we would start getting the same input packet as the output from our module. Although there is a delay in between the input and output packets.