Router top - Overview

The Router 1x3 design follows packet-based protocol and it receives the network packet from a source LAN using **data_in** on a byte-by-byte basis on active posedge of the **clock**. **resetn** is an active low synchronous reset.

The start of a new packet is indicated by asserting **pkt_valid** and the end of the current packet is indicated by de-asserting **pkt_valid**. The design stores the incoming packet inside a FIFO as per the address of the packet. The design has got 3 FIFOs for respective destination LANs.

During packet read operation, the destination LANs monitor **vld_out_x** (x can be 0, 1, or 2), and then assert **read_enb_x** (x can be 0, 1, or 2). The packet is read by the destination LANs using the channels **data_out_x** (x can be 0, 1, or 2).

Sometimes, the router can enter into a busy state which is indicated by the signal **busy.** The busy signal is sent back to the source LAN so that the source has to wait to send the next byte of the packet.

To confirm the correctness of the packet, received by the router, we have implemented an error detection mechanism i.e. parity check. If there is a mismatch in the parity byte sent by the source LAN and the internal parity calculated by the router, then the **error** signal is asserted. This error signal is sent back to the source LAN so that by monitoring the same, the source LAN can resend the packet.

This design can receive only 1 packet at a time, but 3 packets can be read simultaneously.

Router- 1x3 Features

- **Packet Routing:** The packet is driven from the input port and is routed to any one output port, based on the address of the destination network.
- **Parity Checking:** An error detection technique that tests the integrity of digital data being transmitted between Server & Client. This technique ensures that the data transmitted by the Server network is received by the Client network without getting corrupted.
- **Reset:** It is an active low synchronous input that resets the router. Under reset conditions, the router FIFOs are made empty and the valid out signals go low indicating that no valid packet is detected on the output data bus.
- **Sending Packet:** Refer to Router input protocol
- **Reading Packet:** Refer to Router output protocol

Router- Packet

- **Packet Format:** The Packet consists of 3 parts i.e. Header, payload and parity such that each id of 8 bits width and the length of the payload can be extended between 1 byte to 63 bytes.

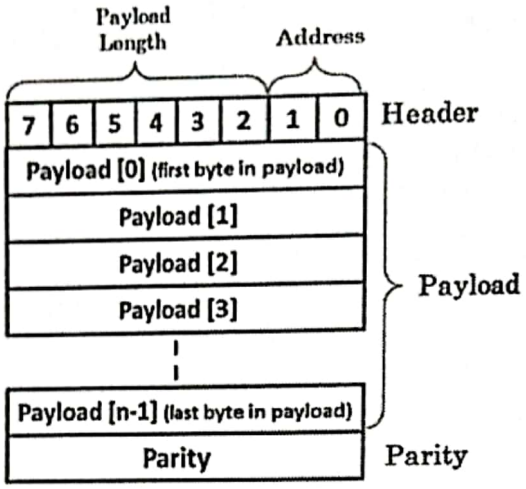
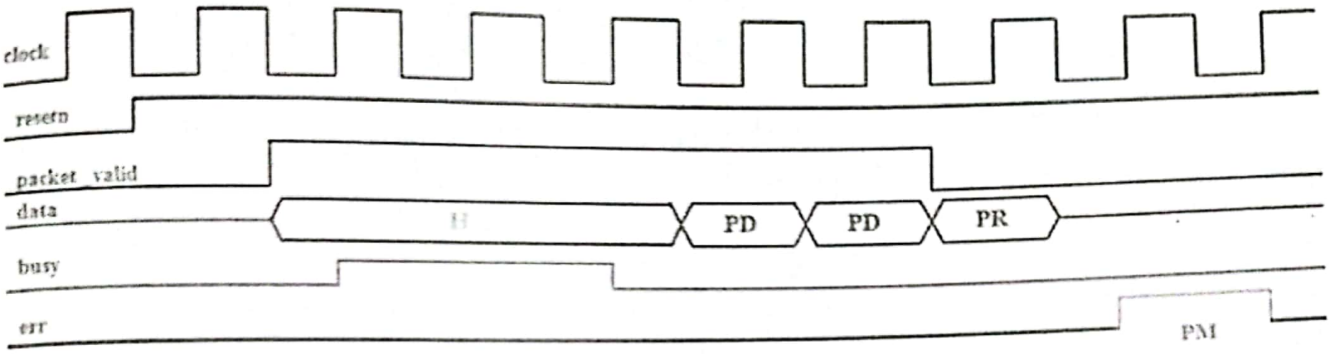


Figure - Packet Format

- **Header: The packet** header contains two fields DA and length.
  - DA: The destination address of the packet is of 2 bits. The router drives the packets to respective ports based on the destination address of the packets. Each output port has a 2-bit unique port address. If the destination address of the packet matches the port address, then the router drives the packet to the output port. The address "3" is invalid.
  - Length: The length of the data is 6 bits. It specifies the number of data bytes. A packet can have a minimum data size of 1 byte and a maximum size of 63 bytes.
    If Length = 1, it means the data length is 1 byte
    If Length = 63, it means the data length is 63 bytes
- **Payload:** Payload is the data information. Data should be in terms of bytes.
- **Parity:** This field contains the security check of the packet. It is calculated as bitwise parity over the header and payload bytes of the packet as mentioned below.

Router- Input Protocol



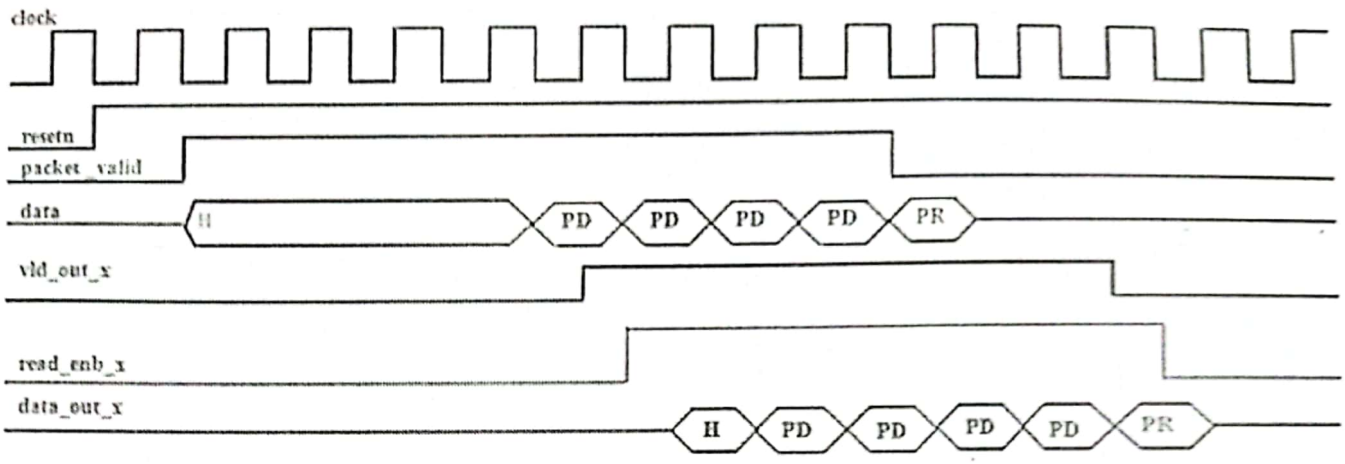
H : Header Byte
PD : Payload Data
PR : Packet Parity byte
PM : Parity Mismatch

The characteristics of the DUT input protocol are as follows:

- **TestBench Note:** All input signals are active high except active low reset and are synchronized to the falling edge of the clock. This is because the DUT router is sensitive to the rising edge of the clock. Therefore, in the testbench, driving input signals on the falling edge ensure adequate setup and hold time. But in the SystemVerilog/UVM-based testbench, the clocking block can be used to drive the signals on the positive edge of the clock itself and thus avoids metastability.
- The packet_valid signal is asserted on the same clock edge when the header byte is driven onto the input data bus.
- Since the header byte contains the address, this tells the router to which output channel the packet should be routed (data_out_0, data_out_1, or data_out_2).
- Each subsequent byte of the payload after the header byte should be driven on the input data bus for every new falling edge of the clock.
- After the last payload byte has been driven, on the next falling clock, the packet_valid signal must be de-asserted, and the packet parity byte should be driven. This signals the completion of the packet.

- The testbench shouldn't drive any bytes when a busy signal is detected instead it should hold the last driven value.
- The "busy" signal when asserted drops any incoming byte of data.
- The "err" signal is asserted when a packet parity mismatch is detected.

Router- Output Protocol



H : Header Byte
PD : Payload Data
PR : Packet Parity byte
x : 0,1,2

The characteristics of the output protocol are as follows:

- **TestBench Note:** All output signals are active high and are synchronized to the rising edge of the clock.
- Each output port data_out_X (data_out_0, data_out_1, data_out_2) is internally buffered by a FIFO of size 16X9.
- The router asserts the vld_out_X.(vld_out_0, vld_out_1 or vld_out_2) signal when valid data appears on the vld_out_X (data_out_0, data_out_1 or data_out_2) output bus. This is a signal to the receiver's client which indicates that data is available on a particular output data bus.
- The packet receiver will then wait until it has enough space to hold the bytes of the packet and then respond with the assertion of the read_enb_X (read_enb_0, read_enb1 or read_enb_2) signal.
- The read_enb_X (read_enb_0, read_enb_1 or read_enb_2) input signal can be asserted on the falling clock edge in which data are read from the data_out_X (data_out_0, data_out_1 or data_out_2) bus.

- The read_enb_X (read_enb_0, read_enb_1 or read_enb_2) must be asserted within 30 clock cycles of vld_out_X (vld_out_0, vld_out_1 or vld_out_2) being asserted else time-out occurs, which resets the FIFO.
- The data_out_X bus will be tri-stated during a scenario when a packet's byte is lost due to a time-out condition.