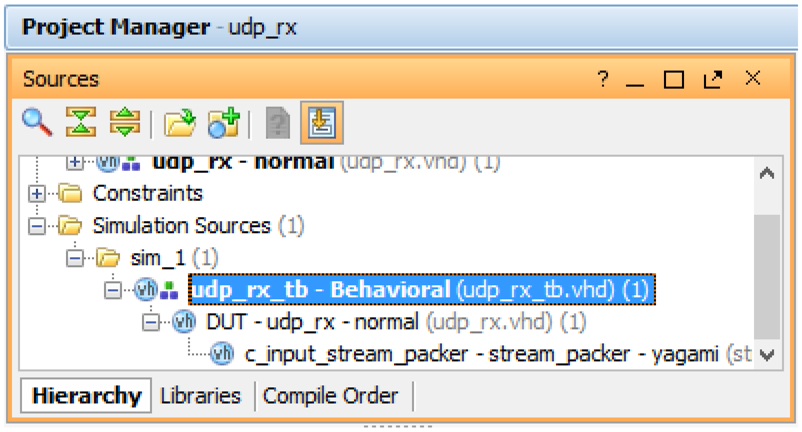
How to run simulation for the UDP Receiver Module

The testbench is designed to read input data from text file and write the output of the module to text file. Each input text file represents test scenario for the receiver module. The steps to run any scenario are as follow:

1. Open udp_rx.xpr/udp_rx_post_synth.xpr project in Vivado and then open the simulation source file.



Project Manager - udp_rx
Sources
udp_rx - normal (udp_rx.vhd) (1)
Constraints
Simulation Sources (1)
sim_1 (1)
udp_rx_tb - Behavioral (udp_rx_tb.vhd) (1)
DUT - udp_rx - normal (udp_rx.vhd) (1)
c_input_stream_packer - stream_packer - yagami (st
Hierarchy
Libraries
Compile Order

2. Scroll down in the testbench to line 59 to change the input file name to which text file you are going to use in the simulation.

```
file In_file : text open read_mode is "zero-length.txt";-- Change the file name
```

3. Scroll down to line 83 to change the number of udp packets will be tested based on the provided table.

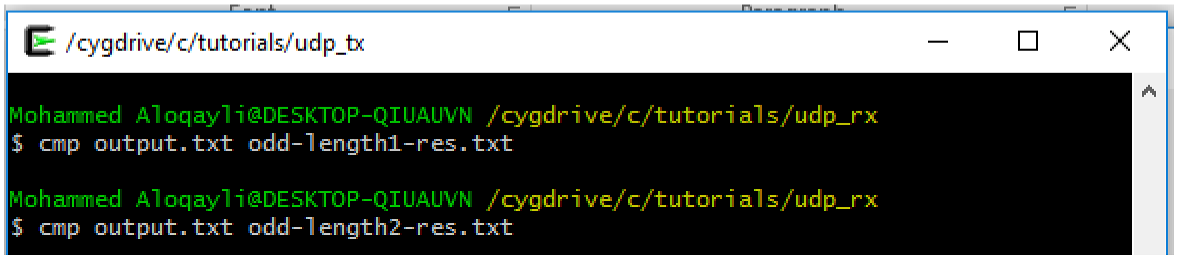
```
signal Num_of_pckts : POSITIVE := 1;
```

Text File	Number of UDP Packets
zero-length.txt	1
odd-length1.txt	1
odd-length2.txt	1
even-length.txt	1
maxudp.txt	1
consecutive-packets.txt	3

4. Run behavioral/post-synthesis simulation for at least 500 ns except for the maximum udp packet test case you need to run the simulation for 100us.
5. Then the testbench will write the output of the module to a text file "output.txt"

```
file Out_file : text open write_mode is "output.txt";-- Change the file name
```

6. In order to check whether the module is behaving as expected or not, you need to compare between "output.txt" file and expected result text file (i.e. zero length test case will have expected result in "zero-length-res.txt").
7. To compare between text files, they should be in the same directory. Then, you need to execute cmp command in Linux. for Windows you can use Cygwin to execute linux commands. If they are matching nothing will be returned in the prompt.



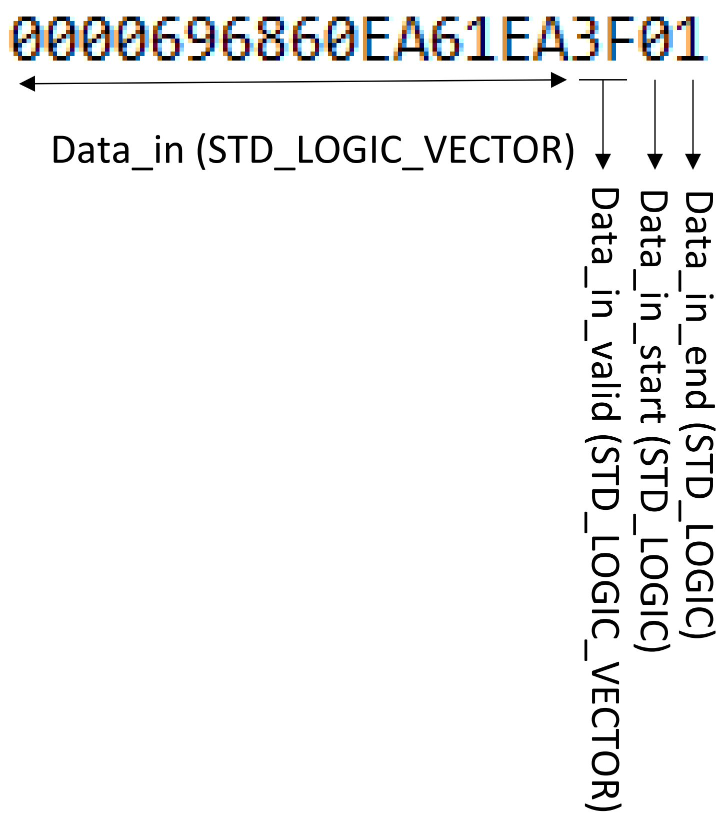
```
Mohammed Aloqayli@DESKTOP-QIUAUVN /cygdrive/c/tutorials/udp_rx
$ cmp output.txt odd-length1-res.txt

Mohammed Aloqayli@DESKTOP-QIUAUVN /cygdrive/c/tutorials/udp_rx
$ cmp output.txt odd-length2-res.txt
```

Test Data Format:

Each input text file is arranged as follow:

- First 8 bytes represent data in hexadecimal.
- Next 2 bytes represent valid signal in hexadecimal.
- Next bit represents start signal in binary.
- Last bit represents end signal in binary.


0000696860EA61EA3F01
Data_in (STD_LOGIC_VECTOR)
Data_in_valid (STD_LOGIC_VECTOR)
Data_in_start (STD_LOGIC)
Data_in_end (STD_LOGIC)