Name: Sushan Patel
UIN: 334003274

Lab 07 Report

1. Driver Code:

```
//////////////////////////////////////////////////////////////////////////////
// Texas A&M University
// CSCE 616 Hardware Design Verification
// Created by  : Prof. Quinn and Saumil Gogri
//////////////////////////////////////////////////////////////////////////////

class htax_tx_driver_c extends uvm_driver #(htax_packet_c);

        parameter PORTS = `PORTS;
        parameter VC            = `VC;
        parameter WIDTH = `WIDTH;

        `uvm_component_utils(htax_tx_driver_c)

        virtual interface htax_tx_interface htax_tx_intf;

        //Constructor
        function new (string name, uvm_component parent);
                super.new(name,parent);
        endfunction : new


 //UVM build phase
 function void build_phase(uvm_phase phase);
  super.build_phase(phase);
                if(!uvm_config_db#(virtual htax_tx_interface)::get(this,"","tx_vif",htax_tx_intf))
                        `uvm_fatal("NO_TX_VIF",{"Virtual Interface needs to be set for ",
get_full_name(),".tx_vif"})
        endfunction : build_phase

        //UVM run_phase
 task run_phase(uvm_phase phase);
                        rst_dut_and_signal();                                   //Reset
the interface signals
                forever begin
                //Till end of test
                        seq_item_port.get_next_item(req);       //Get next pkt from sequencer
                        drive_thru_dut(req);                                    //Drive
pkt to DUT via interface
```

```
                seq_item_port.item_done();                    //Sends ACK to
sequencer
            end
        endtask : run_phase

        task rst_dut_and_signal();
            @(negedge htax_tx_intf.rst_n);
            `uvm_info("TOP", "Resetting the DUT synchronously", UVM_NONE)
            @(posedge htax_tx_intf.clk)
            htax_tx_intf.tx_outport_req  = 'b0;
        htax_tx_intf.tx_vc_req      = 'b0;
        htax_tx_intf.tx_sot         = 'b0;
        htax_tx_intf.tx_release_gnt  = 'b0;
        htax_tx_intf.tx_eot         = 'b0;
        endtask : rst_dut_and_signal

        extern protected task drive_thru_dut(htax_packet_c pkt);
endclass : htax_tx_driver_c

//TO DO : COMPLETE THE DRIVER CODE BELOW
task htax_tx_driver_c::drive_thru_dut(htax_packet_c pkt);

        `uvm_info (get_type_name(), $sformatf("Input Data Packet to DUT : \n%s",
pkt.sprint()),UVM_NONE)

        //Wait for pkt.delay clock cycles
        repeat (pkt.delay) @(posedge htax_tx_intf.clk);

        //On next clk-posedge
        //TO DO : Set htax_tx_intf.tx_outport_req in one-hot fashion from pkt.dest_port
        @(posedge htax_tx_intf.clk)
        case(pkt.dest_port)
            'b00: htax_tx_intf.tx_outport_req <= 4'b0001;
            'b01: htax_tx_intf.tx_outport_req <= 4'b0010;
            'b10: htax_tx_intf.tx_outport_req <= 4'b0100;
            'b11: htax_tx_intf.tx_outport_req <= 4'b1000;
            default: htax_tx_intf.tx_outport_req <= 4'b0001;
        endcase

        //TO DO : Assign htax_tx_intf.tx_vc_req from pkt.vc
        htax_tx_intf.tx_vc_req <= pkt.vc;


        @(posedge htax_tx_intf.clk)
        //TO DO : Wait till htax_tx_intf.tx_vc_gnt is received
        while (~(|htax_tx_intf.tx_vc_gnt)) begin
            @(posedge htax_tx_intf.clk)
            `uvm_info("TOP", "Waiting for tx_vc_gnt", UVM_NONE)
```

```
        end


        //On next clk-posedge
        //TO DO : Set any one bit tx_sot[vc-1:0] to 1 from the ones granted by htax_tx_intf.tx_vc_gnt
        //                                      (refer to SPEC doc for more details)

        for(int i='d0; i < `VC ; i++) begin  //TO DO : Replace XXX and YYY with appropriate values for a packet
                if (htax_tx_intf.tx_vc_gnt[i])
                        htax_tx_intf.tx_sot[i] <= 'b1;
                else
                        htax_tx_intf.tx_sot[i] <= 'b0;
        end
        //TO DO : Drive pkt.data[0] on htax_tx_intf.tx_data
        htax_tx_intf.tx_data <= pkt.data[0];

        //TO DO : Reset htax_tx_intf.tx_outport_req and htax_tx_intf.tx_vc_req (to zero)
        htax_tx_intf.tx_outport_req  <= 'b0;
        htax_tx_intf.tx_vc_req      <= 'b0;

        //TO DO : On consecutive clk-posedges drive each of the packet's pkt.data on htax_tx_intf.tx_data
        //TO DO : Assign htax_tx_intf.tx_sot to zero after first cycle
        //TO DO : Assert htax_tx_intf.tx_release_gnt for one clock cycle when driving second last data packet
        //TO DO : Assert htax_tx_intf.tx_eot for one clock cycle when driving last data packet
        for(int i='d1; i < pkt.length ; i++) begin  //TO DO : Replace XXX and YYY with appropriate values for a packet
                @(posedge htax_tx_intf.clk)
                        htax_tx_intf.tx_data <= pkt.data[i];
                        if (i=='d1) begin
                                htax_tx_intf.tx_sot <= 'b0;
                        end
                        if (i==(pkt.length-'d2)) begin
                                htax_tx_intf.tx_release_gnt <= 'b1;
                        end
                        if (i==(pkt.length-'d1)) begin
                                htax_tx_intf.tx_release_gnt <= 'b0;
                                htax_tx_intf.tx_eot <= 'b1;
                        end
        end

        //On next clk-posedge
        //TO DO : Assign htax_tx_intf.tx_data to X and htax_tx_intf.tx_eot to zero
        @(posedge htax_tx_intf.clk)

        htax_tx_intf.tx_data <= 'bX;
```

```
        htax_tx_intf.tx_eot  <= 'b0;


        `uvm_info (get_type_name(), $sformatf("Ended Driving Data Packet to DUT"), UVM_NONE)
endtask : drive_thru_dut
```

2. Monitor Code:

```
////////////////////////////////////////////////////////////////////////
// Texas A&M University
// CSCE 616 Hardware Design Verification
// Created by  : Prof. Quinn and Saumil Gogri
////////////////////////////////////////////////////////////////////////

class htax_tx_monitor_c extends uvm_monitor;

        parameter PORTS = `PORTS;

        //Factory Registration
        `uvm_component_utils(htax_tx_monitor_c)

        //uvm_analysis_port #(htax_tx_mon_packet_c) tx_collect_port;

        virtual interface htax_tx_interface htax_tx_intf;
        htax_tx_mon_packet_c tx_mon_packet;
        int pkt_len;

        //constructor
        function new (string name, uvm_component parent);
                super.new(name,parent);
                //tx_collect_port = new("tx_collect_port",this);
                tx_mon_packet           = new();
        endfunction : new

 //UVM build phase
 function void build_phase(uvm_phase phase);
  super.build_phase(phase);
                if(!uvm_config_db#(virtual htax_tx_interface)::get(this,"","tx_vif",htax_tx_intf))
                        `uvm_fatal("NO_TX_VIF",{"Virtual Interface needs to be set for ",
get_full_name(),".tx_vif"})
        endfunction : build_phase

        //TO DO : Complete the run_phase tasks with Hints below
        task run_phase(uvm_phase phase);
                forever begin
                        pkt_len=0;

                        @(posedge |htax_tx_intf.tx_vc_gnt) begin
```

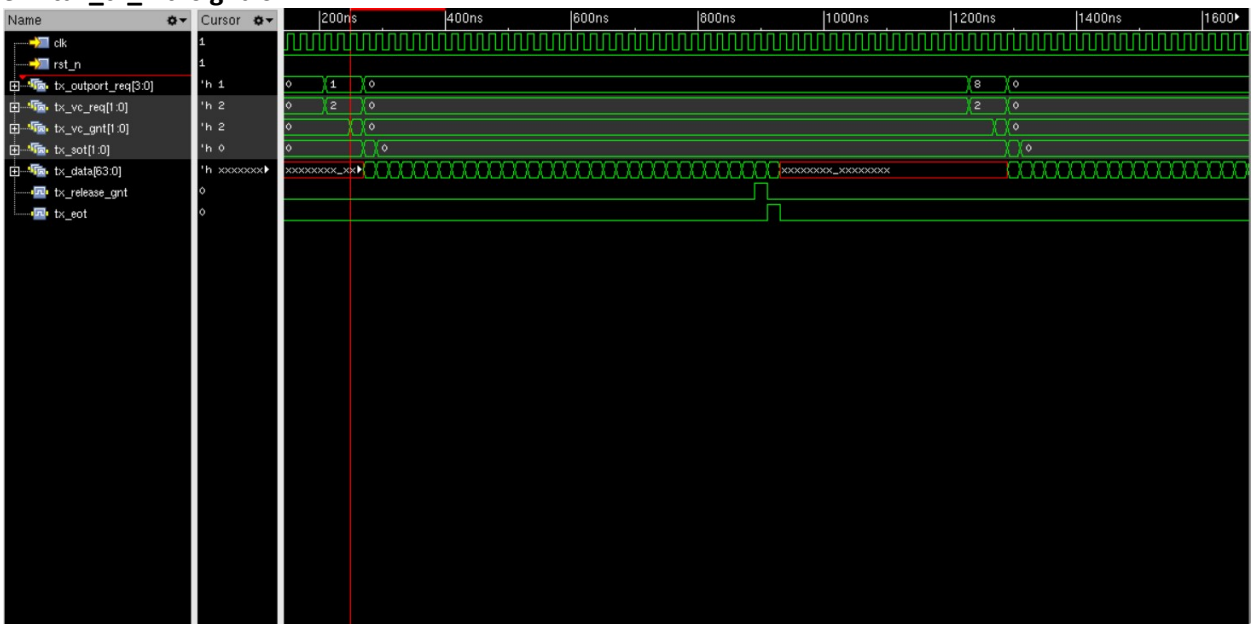
```
                        //TO DO : Assign tx_mon_packet.dest_port from htax_tx_intf.tx_outport_req
                        tx_mon_packet.dest_port <= htax_tx_intf.tx_outport_req;
                        case(htax_tx_intf.tx_outport_req)
                                'b0001: tx_mon_packet.dest_port <= 'b00;
                                'b0010: tx_mon_packet.dest_port <= 'b01;
                                'b0100: tx_mon_packet.dest_port <= 'b10;
                                'b1000: tx_mon_packet.dest_port <= 'b11;
                                default: tx_mon_packet.dest_port = 'b00;
                        endcase
                end

                @(posedge htax_tx_intf.clk);
                //TO DO : On consequtive cycles append htax_tx_intf.tx_data to tx_mon_packet.data[] till htax_tx_intf.tx_eot pulse
                while(~ htax_tx_intf.tx_eot) begin // TO DO : Replace XXX with appropriate condition

                @(posedge htax_tx_intf.clk);
                tx_mon_packet.data = new[++pkt_len](tx_mon_packet.data);
                tx_mon_packet.data[pkt_len-1] =  htax_tx_intf.tx_data;
                end
                tx_mon_packet.print();
        end
    endtask : run_phase

endclass : htax_tx_monitor_c
```

3. Htax_tx_intf signals:



4. Simulation Output:

```
--- UVM Report catcher Summary ---


Number of demoted UVM_FATAL reports  :    0
Number of demoted UVM_ERROR reports  :    0
Number of demoted UVM_WARNING reports:    0
Number of caught UVM_FATAL reports   :    0
Number of caught UVM_ERROR reports   :    0
Number of caught UVM_WARNING reports :    0

--- UVM Report Summary ---

** Report counts by severity
UVM_INFO :   72
UVM_WARNING :    0
UVM_ERROR :    0
UVM_FATAL :    0
** Report counts by id
[RNTST]     1
[TEST_DONE]     1
[TOP]    35
[UVMTOP]     1
[htax_tx_driver_c]    30
[simple_random_seq]     3
[simple_random_test]     1
Simulation complete via $finish(1) at time 13030 NS + 52
/opt/coe/cadence/XCELIUM/tools/methodology/UVM/CDNS-1.1d/sv/src/base/uvm_root.svh:457     $finish;
xcelium> exit

coverage setup:
  workdir  :  ./cov_work
  dutinst  :  top(top)
  scope    :  scope
  testname :  test

coverage files:
  model(design data) :  ./cov_work/scope/icc_1cfdfc8b_4389b1cd.ucm (reused)
  data               :  ./cov_work/scope/test/icc_1cfdfc8b_4389b1cd.ucd
TOOL:   xrun    22.03-s004: Exiting on Oct 31, 2023 at 17:42:28 CDT  (total: 00:00:07)
```

5. HTAX Packet / HTAX TX Monitor Packet:

```
UVM_INFO ../tb/htax_tx_driver_c.sv(57) @ 8050000: uvm_test_top.tb.tx_port[1].tx_driver [htax_tx_driver_c] Input Data Packet to DUT :
----------------------------------------------------------------------------------------------------------
Name                          Type           Size  Value
----------------------------------------------------------------------------------------------------------
req                           htax_packet_c  -     @6295
  delay                       integral       32    'hf
  dest_port                   integral       32    'h1
  vc                          integral       2     'h3
  length                      integral       32    'h7
  data                        da(integral)   7     -
    [0]                       integral       64    'hb6918cdbd903a359
    [1]                       integral       64    'he1b7eda52a2fd37d
    [2]                       integral       64    'h5f40596fd7677d47
    [3]                       integral       64    'h3d17edba28b3ce93
    [4]                       integral       64    'h18b3aecf13b55b7e
    [5]                       integral       64    'hfd73d0607a25b643
    [6]                       integral       64    'h42c410dd7f935b23
  begin_time                  time           64    8050000
  depth                       int            32    'd2
  parent sequence (name)      string         17    simple_random_seq
  parent sequence (full name) string         54    uvm_test_top.tb.tx_port[1].sequencer.simple_random_seq
  sequencer                   string         36    uvm_test_top.tb.tx_port[1].sequencer
----------------------------------------------------------------------------------------------------------

UVM_INFO ../tb/htax_tx_driver_c.sv(81) @ 8410000: uvm_test_top.tb.tx_port[1].tx_driver [TOP] Waiting for tx_vc_gnt
UVM_INFO ../tb/htax_tx_driver_c.sv(81) @ 8430000: uvm_test_top.tb.tx_port[1].tx_driver [TOP] Waiting for tx_vc_gnt
UVM_INFO ../tb/htax_tx_driver_c.sv(129) @ 8570000: uvm_test_top.tb.tx_port[1].tx_driver [htax_tx_driver_c] Ended Driving Data Packet to DUT
----------------------------------------------------------------------
Name                 Type                 Size  Value
----------------------------------------------------------------------
htax_tx_mon_packet_c  htax_tx_mon_packet_c  -     @4568
  dest_port          integral             32    'h1
  data               da(integral)         7     -
    [0]              integral             64    'hb6918cdbd903a359
    [1]              integral             64    'he1b7eda52a2fd37d
    [2]              integral             64    'h5f40596fd7677d47
    [3]              integral             64    'h3d17edba28b3ce93
    [4]              integral             64    'h18b3aecf13b55b7e
    [5]              integral             64    'hfd73d0607a25b643
    [6]              integral             64    'h42c410dd7f935b23
----------------------------------------------------------------------
UVM_INFO ../tb/htax_tx_driver_c.sv(57) @ 8570000: uvm_test_top.tb.tx_port[1].tx_driver [htax_tx_driver_c] Input Data Packet to DUT :
```