



Crack DV Interviews at Top Product Companies

NVIDIA
SYNOPSYS
QUALCOMM
AMD
INTEL

Top 50 Most Frequently
Asked Interview Questions

1. Write a constraint to generate only odd numbers between 5 and 15, excluding any consecutive odd numbers

Company: Intel, Synopsys

```
class my_rand;
    rand bit [3:0] a;
```

```
constraint c1 {
    a inside {[5:15]};
    a % 2 == 1;
    !(a == 7 || a == 9 || a == 11 || a == 13); // remove consecutive odd
}
endclass
```

```
module tb;
    my_rand obj = new();
    initial begin
        repeat(10) begin
            void'(obj.randomize());
            $display("a = %0d", obj.a);
        end
    end
endmodule
```

2. Implement polymorphism using base and derived classes in SystemVerilog

Company: Qualcomm, AMD

```
class Packet;
    virtual function void send();
        $display("Sending base packet");
    endfunction
endclass
```

```
class EthernetPacket extends Packet;
    function void send();
        $display("Sending Ethernet packet");
    endfunction
endclass
```

```

endfunction
endclass

module tb;
  Packet pkt;
  initial begin
    pkt = new(); pkt.send();
    pkt = new EthernetPacket(); pkt.send(); // Polymorphism
  end
endmodule

```

3. Create and sample a covergroup inside a class

Company: Intel, Apple

```

class transaction;
  rand bit [3:0] data;

  covergroup cg;
    option.per_instance = 1;
    DATA: coverpoint data {
      bins low = {[0:7]};
      bins high = {[8:15]};
    }
  endgroup

  function new();
    cg = new();
  endfunction

  function void sample_data();
    cg.sample();
  endfunction
endclass

```

```

module tb;
  transaction t;

```

```

initial begin
    t = new();
    repeat(10) begin
        void'(t.randomize());
        t.sample_data();
        $display("Data = %0d", t.data);
    end
end
endmodule

```

4. Build a simple scoreboard using TLM analysis FIFO and analysis export

Company: Qualcomm, Synopsys

```

class transaction;
    rand bit [3:0] data;
    bit [7:0] id;
endclass

```

```

class scoreboard extends uvm_component;
    uvm_analysis_imp #(transaction, scoreboard) analysis_export;
    uvm_tlm_analysis_fifo #(transaction) fifo;

```

```

function new(string name, uvm_component parent);
    super.new(name, parent);
    analysis_export = new("analysis_export", this);
    fifo = new("fifo", this);
endfunction

```

```

virtual function void write(transaction t);
    fifo.write(t);
endfunction

```

```

task run_phase(uvm_phase phase);
    transaction exp, act;
    forever begin

```

```

fifo.get(act);
  exp = get_expected(act.id); // assume get_expected() is user-
defined
  if (act.data !== exp.data)
    `uvm_error("SCOREBOARD", $sformatf("Mismatch: ID=%0d
Expected=%0d Actual=%0d", act.id, exp.data, act.data))
  end
endtask
endclass

```

5. Write a virtual sequence that starts two sequences on two different agents concurrently

Company: Intel, AMD

```

class my_virtual_seq extends uvm_sequence #
(uvm_sequence_item);
`uvm_object_utils(my_virtual_seq)

my_seq_agent1 seq1;
my_seq_agent2 seq2;

virtual_seq_if vif1, vif2;

function new(string name = "my_virtual_seq");
  super.new(name);
endfunction

task body();
  seq1 = my_seq_agent1::type_id::create("seq1");
  seq2 = my_seq_agent2::type_id::create("seq2");
  fork
    seq1.start(env.agent1.sequencer);
    seq2.start(env.agent2.sequencer);
  join
endtask
endclass

```

6. How do you override a component in UVM using the factory mechanism?

Company: Nvidia, Synopsys

```
class monitor extends uvm_component;
`uvm_component_utils(monitor)
function new(string name, uvm_component parent);
super.new(name, parent);
endfunction
endclass
```

```
class monitor_ovrd extends monitor;
`uvm_component_utils(monitor_ovrd)
function new(string name, uvm_component parent);
super.new(name, parent);
endfunction
endclass
```

```
class my_test extends uvm_test;
`uvm_component_utils(my_test)
function new(string name, uvm_component parent);
super.new(name, parent);
endfunction
```

```
function void build_phase(uvm_phase phase);
factory.set_type_override_by_type(monitor::get_type(),
monitor_ovrd::get_type());
super.build_phase(phase);
endfunction
endclass
```

7. Access and verify a register using UVM register model (uvm_reg)

```
task test_reg_model();
    uvm_status_e status;
    uvm_reg_data_t data;
```

Company: Apple, AMD

```
// write to register
reg_model.ctrl_reg.write(status, 'hA5);

// mirror (read and compare)
reg_model.ctrl_reg.mirror(status, UVM_CHECK);
```

```
// predictive update
reg_model.ctrl_reg.predict('hA5);
endtask
```

8. How do you pass a config object between UVM components using uvm_config_db?

Company: Intel, Qualcomm

```
// In test
task build_phase(uvm_phase phase);
    super.build_phase(phase);
    my_config cfg = new();
    uvm_config_db #(my_config)::set(this, "*", "cfg", cfg);
endtask
```

```
// In driver
task build_phase(uvm_phase phase);
    super.build_phase(phase);
    if (!uvm_config_db #(my_config)::get(this, "", "cfg", cfg))
        `uvm_fatal("CFG", "Failed to get config")
endtask
```

9. Randomize an array such that it contains only even numbers between 10 and 50, without duplicates

Company: Qualcomm, Apple

```

class rand_array;
  rand bit [7:0] arr[10];

  constraint c_even {
    foreach (arr[i])
      arr[i] inside {[10:50]} && arr[i] % 2 == 0;
  }

  constraint c_unique {
    unique {arr};
  }
endclass

module tb;
  rand_array obj = new();
  initial begin
    if (obj.randomize())
      $display("Randomized Array = %p", obj.arr);
    else
      $display("Randomization Failed");
  end
endmodule

```

10. How do you implement callback mechanism in UVM to inject behavior without modifying driver?

Company: Synopsys, NVIDIA

```

class my_callback extends uvm_callback;
  `uvm_object_utils(my_callback)
  virtual task pre_drive();
    `uvm_info("CALLBACK", "Inside pre_drive callback", UVM_LOW)
  endtask
endclass

```

```

class my_driver extends uvm_driver #(my_transaction);
`uvm_component_utils(my_driver)



virtual task run_phase(uvm_phase phase);
  my_transaction tr;
  forever begin
    seq_item_port.get_next_item(tr);

    // Invoke callback
    uvm_callbacks#(my_driver, my_callback)::invoke("pre_drive",
this);

    // Drive the item
    $display("Driving: %0d", tr.data);
    seq_item_port.item_done();
  end
endtask
endclass

```

11. How to write a covergroup that samples a random transaction and bins specific ranges?

Company: AMD, Intel

```

class tx;
  rand bit [7:0] addr;

  covergroup addr_cg;
    coverpoint addr {
      bins low = {[0:63]};
      bins high = {[64:127]};
    }
  endgroup

  function new();

```

```

addr_cg = new();
endfunction
endclass

module tb;
tx t = new();
initial begin
repeat(10) begin
void'(t.randomize());
t.addr_cg.sample();
$display("Random Addr = %0d", t.addr);
end
end
endmodule

```

12. Override a sequence item using the factory and show polymorphism

Company: Apple, Qualcomm

```

class base_tx extends uvm_sequence_item;
rand bit [7:0] data;
`uvm_object_utils(base_tx)
endclass

class new_tx extends base_tx;
`uvm_object_utils(new_tx)
constraint data_c { data inside {[0:127]}; }
endclass

initial begin
uvm_factory::get().set_type_override_by_type(base_tx::get_type(),
new_tx::get_type());

base_tx tx = base_tx::type_id::create("tx");
void'(tx.randomize());
$display("Polymorphic Data: %0d", tx.data);
end

```

13. Create a UVM sequence that generates and sends a stream of packets with delay between each

Company: Intel, Synopsys

```
class stream_seq extends uvm_sequence #(my_transaction);
`uvm_object_utils(stream_seq)

task body();
    my_transaction tx;
    repeat (5) begin
        tx = my_transaction::type_id::create("tx");
        void'(tx.randomize());
        start_item(tx);
        finish_item(tx);
        `uvm_info("SEQ", $sformatf("Sent tx with data=%0d", tx.data),
UVM_MEDIUM)
        #10ns;
    end
endtask
endclass
```

14. How to add a register field mirroring and prediction in UVM RAL?

Company: AMD, Synopsys

```
task test_reg();
    uvm_status_e status;
    uvm_reg_data_t read_val;

    // Write and predict
    regmodel.status_reg.write(status, 8'h55);
    regmodel.status_reg.predict(8'h55);

    // Mirror and compare
    regmodel.status_reg.mirror(status, UVM_CHECK);
endtask
```

15. Write a SystemVerilog constraint to randomize 2 variables such that sum is always even

Company: Qualcomm, Intel

```

class sum_even;
rand bit [3:0] a, b;

constraint even_sum { (a + b) % 2 == 0; }

endclass

module tb;
    sum_even obj = new();
    initial begin
        repeat (5) begin
            void'(obj.randomize());
            $display("a=%0d, b=%0d, sum=%0d", obj.a, obj.b, obj.a + obj.b);
        end
    end
endmodule

```

16. How to use uvm_config_db to pass a handle from test to environment?

Company: Apple, Synopsys

```

class my_agent_config extends uvm_object;
    `uvm_object_utils(my_agent_config)

    bit enable_monitor;

    function new(string name = "my_agent_config");
        super.new(name);
    endfunction
endclass

class my_agent extends uvm_agent;
    `uvm_component_utils(my_agent)

    my_agent_config cfg;

```

```

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

function void build_phase(uvm_phase phase);
  super.build_phase(phase);
  if (!uvm_config_db #(my_agent_config)::get(this, "", "cfg", cfg))
    `uvm_fatal("CFG_MISSING", "Agent config not found")
endfunction
endclass

class my_test extends uvm_test;
`uvm_component_utils(my_test)

function void build_phase(uvm_phase phase);
  super.build_phase(phase);

  my_agent_config cfg = new();
  cfg.enable_monitor = 1;

  uvm_config_db #(my_agent_config)::set(this, "*", "cfg", cfg);
endfunction
endclass

```

17. FSM to detect 3 consecutive 1s in a serial bit stream

Company: Intel, Qualcomm

```

module detect_3ones(
  input logic clk, rst, in_bit,
  output logic detect
);
  typedef enum logic [1:0] {IDLE, ONE1, ONE2, THREE_ONES} state_t;
  state_t state, next;

  always_ff @ (posedge clk or posedge rst)

```

```

if (rst) state <= IDLE;
else    state <= next;

always_comb begin
  next = IDLE;
  case (state)
    IDLE:    next = in_bit ? ONE1 : IDLE;
    ONE1:   next = in_bit ? ONE2 : IDLE;
    ONE2:   next = in_bit ? THREE_ONES : IDLE;
    THREE_ONES: next = in_bit ? THREE_ONES : IDLE;
  endcase
end

assign detect = (state == ONE2 && in_bit);
endmodule

```

18. Create a coverage model to track burst length patterns in a protocol

Company: AMD, Apple

```

class bus_tx;
  rand bit [3:0] burst_len;

  covergroup burst_cg;
    coverpoint burst_len {
      bins single = {1};
      bins short  = {[2:4]};
      bins long   = {[5:15]};
    }
  endgroup

  function new();
    burst_cg = new();
  endfunction

endclass

module tb;
  bus_tx tx = new();

```

```

initial begin
repeat (10) begin
    void'(tx.randomize());
    tx.burst_cg.sample();
    $display("Burst Length = %0d", tx.burst_len);
end
end
endmodule

```

19. Implement a layered sequence (one sequence calls another)

Company: Qualcomm, Synopsys

```

class sub_seq extends uvm_sequence #(my_transaction);
`uvm_object_utils(sub_seq)

task body();
    my_transaction tx = my_transaction::type_id::create("tx");
    void'(tx.randomize());
    start_item(tx);
    finish_item(tx);
    `uvm_info("SUB_SEQ", $sformatf("TX data: %0d", tx.data),
UVM_LOW)
endtask
endclass

```

```

class main_seq extends uvm_sequence #(my_transaction);
`uvm_object_utils(main_seq)

task body();
repeat (3) begin
    sub_seq s = sub_seq::type_id::create("sub_seq");
    s.start(m_sequencer);
end
endtask
endclass

```

20. Write a constraint for sorting a 4-element array in ascending order

Company: Apple, Synopsys

```
class sort_array;
    rand bit [7:0] arr[4];

    constraint c_sorted {
        arr[0] <= arr[1];
        arr[1] <= arr[2];
        arr[2] <= arr[3];
    }
endclass

module tb;
    sort_array s = new();
    initial begin
        if (s.randomize())
            $display("Sorted Array = %p", s.arr);
    end
endmodule
```

21. Implement UVM objection to end simulation after 10 transactions

Company: AMD, Intel

```
class my_env extends uvm_env;
    `uvm_component_utils(my_env)

    int tx_count;

    function void build_phase(uvm_phase phase);
        super.build_phase(phase);
        tx_count = 0;
    endfunction

    task run_phase(uvm_phase phase);
```

```

phase.raise_objection(this);
repeat (10) begin
    // assume tx is completed
    #10ns;
    tx_count++;
    `uvm_info("ENV", $sformatf("Transaction %0d done", tx_count),
UVM_LOW)
end
phase.drop_objection(this);
endtask
endclass

```

22. Constraint two variables such that one is twice the other and their sum is < 50.

Company: Apple, Synopsys

```

class rel_constraint;
    rand bit [7:0] a, b;

    constraint twice {
        a == 2 * b;
        a + b < 50;
    }
endclass

module tb;
    rel_constraint obj = new();
initial begin
    repeat (5) begin
        void'(obj.randomize());
        $display("a=%0d, b=%0d, sum=%0d", obj.a, obj.b, obj.a + obj.b);
    end
end
endmodule

```

23. How to override a component using uvm_factory?

Company: Qualcomm, Synopsys

```
class base_monitor extends uvm_monitor;
  `uvm_component_utils(base_monitor)
  function new(string name, uvm_component parent);
    super.new(name, parent); endfunction
    function void build_phase(uvm_phase phase);
      super.build_phase(phase);
      `uvm_info("BASE_MON", "Inside base monitor", UVM_LOW)
    endfunction
endclass

class my_monitor extends base_monitor;
  `uvm_component_utils(my_monitor)
  function void build_phase(uvm_phase phase);
    super.build_phase(phase);
    `uvm_info("MY_MON", "Inside overridden monitor", UVM_LOW)
  endfunction
endclass

class my_env extends uvm_env;
  `uvm_component_utils(my_env)
  function void build_phase(uvm_phase phase);
    super.build_phase(phase);
    base_monitor mon = base_monitor::type_id::create("mon", this);
  endfunction
endclass

class my_test extends uvm_test;
  `uvm_component_utils(my_test)
  function void build_phase(uvm_phase phase);
    set_type_override_by_type(base_monitor::get_type(),
    my_monitor::get_type());
    super.build_phase(phase);
  endfunction
endclass
```

24. Write an assertion to check that req stays high for at least 3 cycles

Company: Intel, Apple

// Inside an interface or module with clk, req

```
property p_req_min_hold;
  @(posedge clk) req |=> req[*2]; // 3 cycles total
endproperty

assert property (p_req_min_hold)
  else $fatal("req dropped before 3 cycles!");
```

25. How do you randomize a queue such that it contains unique values from 0–9?

Company: AMD, Qualcomm

```
class unique_q;
  rand bit [3:0] q[$];

  constraint c1 {
    q.size() == 10;
    foreach (q[i], q[j]) i != j -> q[i] != q[j];
    foreach (q[i]) q[i] inside {[0:9]};
  }
endclass

module tb;
  unique_q obj = new();
  initial begin
    if (obj.randomize()) $display("Random Queue: %p", obj.q);
  end
endmodule
```

26. How to connect virtual interfaces in UVM using config_db?

Company: Synopsys, Apple

```
// Interface
interface my_if(input logic clk);
    logic req, ack;
endinterface

// Testbench Top
module tb;
    logic clk = 0;
    always #5 clk = ~clk;

    my_if vif(clk);

    initial begin
        uvm_config_db #(virtual my_if)::set(null, "*", "vif", vif);
        run_test("my_test");
    end
endmodule

// Driver
class my_driver extends uvm_driver #(my_transaction);
    `uvm_component_utils(my_driver)
    virtual my_if vif;

    function void build_phase(uvm_phase phase);
        super.build_phase(phase);
        if (!uvm_config_db #(virtual my_if)::get(this, "", "vif", vif))
            `uvm_fatal("VIF", "Virtual interface not set")
    endfunction
endclass
```

27. Write a class to constrain sum of array elements to 100

Company: Qualcomm, Apple

```
class array_sum;
    rand bit [7:0] arr[5];

    constraint sum_c {
        (arr.sum()) == 100;
    }
endclass

module tb;
    array_sum obj = new();
    initial begin
        repeat (5) begin
            void'(obj.randomize());
            $display("Array = %p, Sum = %0d", obj.arr, obj.arr.sum());
        end
    end
endmodule
```

28. How to use TLM analysis port to send data from monitor to scoreboard?

Company: Intel, Synopsys

```
// Monitor
class my_monitor extends uvm_monitor;
    `uvm_component_utils(my_monitor)

    uvm_analysis_port #(my_transaction) mon_ap;

    function new(string name, uvm_component parent);
        super.new(name, parent);
        mon_ap = new("mon_ap", this);
    endfunction
```

```

task run_phase(uvm_phase phase);
    my_transaction tx = my_transaction::type_id::create("tx");
    // capture tx and send
    mon_ap.write(tx);
endtask
endclass

// Scoreboard
class my_scoreboard extends uvm_scoreboard;
`uvm_component_utils(my_scoreboard)

function void write(my_transaction tx);
`uvm_info("SCOREBOARD", $sformatf("Received tx: %0p", tx),
UVM_LOW)
endfunction
endclass

```

29. Write an OOP-based UVM transaction with deep copy constructor

Company: Apple, AMD

```

class packet extends uvm_sequence_item;
`uvm_object_utils(packet)

rand bit [7:0] data[];
rand int length;

function new(string name = "packet");
    super.new(name);
endfunction

function void do_copy(uvm_object rhs);
    packet p;
    if (!$cast(p, rhs)) return;
    this.length = p.length;
    this.data = new[p.data.size()];
    foreach (p.data[i]) this.data[i] = p.data[i];
endfunction
endclass

```

30. How to use layered virtual sequences in UVM with multiple sequencers?

Company: Qualcomm, Intel

```
class top_seq extends uvm_sequence #(uvm_sequence_item);
`uvm_object_utils(top_seq)

`uvm_declare_p_sequencer(my_virtual_sequencer)

task body();
    seq1 s1 = seq1::type_id::create("s1");
    seq2 s2 = seq2::type_id::create("s2");

    s1.start(p_sequencer.seqr1);
    s2.start(p_sequencer.seqr2);
endtask

endclass
```

```
class my_virtual_sequencer extends uvm_sequencer;
`uvm_component_utils(my_virtual_sequencer)

my_seqr1 seqr1;
my_seqr2 seqr2;

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

endclass
```

31. How to control sequence execution order using arbitration (priority vs. lock)?

Company: Qualcomm, Intel

```
class high_seq extends uvm_sequence;
`uvm_object_utils(high_seq)

task body();
`uvm_info("SEQ", "High priority sequence running", UVM_LOW)
endtask

endclass

class low_seq extends uvm_sequence;
`uvm_object_utils(low_seq)

task body();
`uvm_info("SEQ", "Low priority sequence running", UVM_LOW)
endtask

endclass

class arb_test extends uvm_test;
`uvm_component_utils(arb_test)

function void run_phase(uvm_phase phase);
high_seq h = high_seq::type_id::create("h");
low_seq l = low_seq::type_id::create("l");

fork
begin
h.set_priority(200);
h.start(seqr); // Higher priority
end
begin
l.set_priority(100);
l.start(seqr);
end
join
endfunction

endclass
```

32. Create a covergroup to check all bit transitions of a 2-bit signal

Company: AMD, Apple

```
class transition_cov;
    bit [1:0] sig;

    covergroup cg @(posedge clk);
        coverpoint sig {
            bins all_vals[] = {[0:3]};
            bins transitions = (0=>1=>2=>3);
        }
    endgroup

    function new;
        cg = new();
    endfunction

endclass
```

33. Write a mailbox-based producer-consumer model in SystemVerilog

Company: Synopsys, Qualcomm

```
class ctrl_reg extends uvm_reg;
    rand uvm_reg_field en, mode;

    `uvm_object_utils(ctrl_reg)

    function new(string name = "ctrl_reg");
        super.new(name, 16, UVM_NO_COVERAGE);
    endfunction

    virtual function void build();
        en = uvm_reg_field::type_id::create("en");
        mode = uvm_reg_field::type_id::create("mode");

        en.configure(this, 1, 0, "RW", 0, 1, 1, 0, 0);
        mode.configure(this, 3, 1, "RW", 0, 1, 1, 0, 0);
    endfunction

endclass
```

```
mbox.get(p);
$display("Consumed: %0d", p.data);
endtask
```

```
initial begin
  fork
    producer();
    consumer();
  join
end
```

34. How to define a UVM register model for a 16-bit control register?

Company: Apple, AMD

```
class packet extends uvm_sequence_item;
  `uvm_object_utils(packet)

  rand bit [7:0] data[];
  rand int length;

  function new(string name = "packet");
    super.new(name);
  endfunction

  function void do_copy(uvm_object rhs);
    packet p;
    if (!$cast(p, rhs)) return;
    this.length = p.length;
    this.data = new[p.data.size()];
    foreach (p.data[i]) this.data[i] = p.data[i];
  endfunction

endclass
```

35. Write a SystemVerilog constraint to generate even numbers in a queue

Company: Intel, Apple

```
class even_q;
    rand bit [7:0] q[$];

constraint c1 {
    q.size() == 5;
    foreach (q[i]) q[i] % 2 == 0;
}
endclass
```

36. Explain objection mechanism in UVM with code

Company: Synopsys, Qualcomm

```
task run_phase(uvm_phase phase);
    phase.raise_objection(this);
repeat (5) begin
    `uvm_info("RUN", "Running...", UVM_LOW)
    #10;
end
phase.drop_objection(this);
endtask
```

37. Write a covergroup that tracks transitions from IDLE -> ACTIVE -> DONE

Company: Apple, AMD

```
typedef enum {IDLE, ACTIVE, DONE} state_t;
state_t state;
```

```
covergroup cg @(posedge clk);
    coverpoint state {
        bins full_path = (IDLE => ACTIVE => DONE);
    }
endgroup
```

38. How to randomize a 2D array in SystemVerilog with each row sorted?

Company: Intel, Synopsys

```
class array2D;
    rand bit [7:0] arr[3][4];

    constraint row_sorted {
        foreach (arr[i]) {
            foreach (arr[i,j]) if (j < 3) arr[i][j] <= arr[i][j+1];
        }
    }
endclass
```

39. Write a SystemVerilog constraint to generate unique random values in an array

Company: Synopsys, Apple

```
class unique_array;
    rand bit [3:0] a[5];

    constraint uniq_c { unique { a }; }
endclass
```

```
// Testbench
initial begin
    unique_array u = new();
    if (u.randomize())
        $display("Unique values: %p", u.a);
end
```

40. How to use UVM callbacks in a driver to modify behavior dynamically?

Company: Intel, Qualcomm

```

class my_driver_callback extends uvm_callback;
  virtual task modify_item(my_transaction t);
    t.data = 8'hAA; // Override with fixed pattern
  endtask
endclass

class my_driver extends uvm_driver #(my_transaction);
  `uvm_register_cb(my_driver, my_driver_callback)

  virtual task run_phase(uvm_phase phase);
    forever begin
      seq_item_port.get_next_item(req);
      `uvm_do_callbacks(my_driver, my_driver_callback,
modify_item(req))
      drive(req);
      seq_item_port.item_done();
    end
  endtask
endclass

```

41. How to randomize only specific fields of a transaction in SV?

Company: Synopsys, Apple

```

class pkt;
  rand bit [3:0] header;
  rand bit [7:0] payload;
endclass

initial begin
  pkt p = new();
  p.header = 4'hF;
  p.randomize() with { payload > 10; } // header remains unchanged
  $display("Header = %0h, Payload = %0h", p.header, p.payload);
end

```

42. Implement a UVM monitor that samples packets on every positive clock

Company: Intel, AMD

```
class my_monitor extends uvm_monitor;
    virtual interface vif;
    my_transaction trans;

    `uvm_component_utils(my_monitor)

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

    virtual task run_phase(uvm_phase phase);
        forever begin
            @(posedge vif.clk);
            if (vif.valid) begin
                trans = my_transaction::type_id::create("trans");
                trans.data = vif.data;
                `uvm_info("MON", $sformatf("Sampled data = %0h", trans.data),
                UVM_LOW)
            end
        end
    endtask
endclass
```

43. How to model a scoreboard using TLM analysis port?

Company: Intel, Synopsys

```
class my_scoreboard extends uvm_scoreboard;
`uvm_component_utils(my_scoreboard)

uvm_analysis_imp #(my_transaction, my_scoreboard) ap;

function new(string name, uvm_component parent);
super.new(name, parent);
ap = new("ap", this);
endfunction

function void write(my_transaction t);
`uvm_info("SCB", $sformatf("Received: %0h", t.data), UVM_LOW)
// Compare with expected result
endfunction
endclass
```

44. What is the difference between randcase and case?

Company: Apple, Qualcomm

// randcase - probability-based branching

randcase

70: \$display("Path A"); // 70% chance

30: \$display("Path B"); // 30% chance

endcase

// case - deterministic branching

case(sel)

2'b00: \$display("Path A");

2'b01: \$display("Path B");

endcase

45. Write a SystemVerilog queue constraint such that each element is 2x the previous

Company: Intel, Synopsys

```
class exp_queue;
    rand bit [7:0] q[5];

    constraint exp {
        q[0] == 1;
        foreach(q[i]) if (i > 0) q[i] == 2 * q[i-1];
    }
endclass
```

46. Explain factory override in UVM with code example

Company: Apple, Qualcomm

```
class base_seq extends uvm_sequence;
    `uvm_object_utils(base_seq)
endclass
```

```
class override_seq extends base_seq;
    `uvm_object_utils(override_seq)
endclass
```

```
initial begin
    factory.set_type_override_by_type(base_seq::get_type(),
    override_seq::get_type());
```

```
base_seq b = base_seq::type_id::create("b");
$display("Created object type: %s", b.get_type_name()); // Prints
override_seq
end
```

47. Create an assertion to check that once start is high, ack must go high within 3 cycles

Company: Intel, AMD

```
property ack_within_3;  
  @(posedge clk) start |=> ##[1:3] ack;  
endproperty
```

```
assert property (ack_within_3);
```

48. How to connect multiple sequences to a virtual sequencer in UVM?

Company: Intel, Qualcomm

```
// Virtual sequencer  
class virt_sequencer extends uvm_sequencer;  
  `uvm_component_utils(virt_sequencer)  
  
  my_seqr_1 seqr1;  
  my_seqr_2 seqr2;  
  
  function new(string name, uvm_component parent);  
    super.new(name, parent);  
  endfunction  
endclass  
  
// Virtual sequence  
class virt_sequence extends uvm_sequence;  
  `uvm_object_utils(virt_sequence)  
  
  virt_sequencer vsqr;  
  
  task body();  
    my_seq_1 s1 = my_seq_1::type_id::create("s1");  
    my_seq_2 s2 = my_seq_2::type_id::create("s2");  
  
    s1.start(vsqr.seqr1);  
    s2.start(vsqr.seqr2);  
  endtask  
endclass
```

49. Write a SystemVerilog assertion to ensure req is high for exactly 2 cycles only

Company: Intel, Apple

```
property req_2_cycle;  
  @(posedge clk) $rose(req) |-> req ##1 req ##1 !req;  
endproperty  
  
assert property (req_2_cycle);
```

50. How to create a user-defined UVM objection for custom simulation control?

Company: AMD, Synopsys

```
class my_component extends uvm_component;  
  `uvm_component_utils(my_component)  
  
  function new(string name, uvm_component parent);  
    super.new(name, parent);  
  endfunction  
  
  virtual task run_phase(uvm_phase phase);  
    phase.raise_objection(this, "Start processing");  
    // Perform tasks  
    #100ns;  
    phase.drop_objection(this, "End processing");  
  endtask  
endclass
```



Excellence in World class VLSI Training & Placements

Do follow for updates & enquires



+91- 9182280927