DAY-4

#100DAYSRTL

“Aim”:- To Verify the Clock divider (by clk/odd) using System Verilog Layered Testbenches

“System Verilog Testbenches”:-

Design:-

```
//Author:- Chinna Venkata Narayana Reddy
//Date:-28/08/2023
module ClockDividerBy3n(
    input rst, clk,
    output fby3,fby5,fby9
);
reg [3:0] count,dout;
    always @(posedge clk or posedge rst) begin
        if (rst)
            count <= 0;
        else begin
            if (count == 4'd15)
                count <= 0;
            else
                count <= count + 1;
        end
        end
    always @(negedge clk) begin
    if(rst)
     dout<=0;
    else
     dout<=count;
    end
    assign fby3=count[0]||dout[0];
    assign fby5=count[1]||dout[1];
    assign fby9=count[2]||dout[2];
endmodule
```

Interface:-

```
interface Clk_Div_odd_if;
  logic rst,clk;
  logic div3,div5,div9;
endinterface
```

Transaction:-

```
class transaction;
  rand bit rst;
  bit div3,div5,div9;
  function void display(input string name);
    $display("rst=%b,div3=%b,div5=%b,div9=%b",rst,div3,div5,div9);
  endfunction
  function transaction copy();
    copy=new();
    copy.rst=this.rst;
    copy.div3=this.div3;
    copy.div5=this.div5;
    copy.div9=this.div9;
  endfunction
endclass
```

Generator:-

```
class generator;
  transaction tr;
  int count;
  event sconext;
  event done;
  mailbox (transaction) mbx;
  mailbox (transaction) mbxref;
  function new(mailbox (transaction) mbx,mailbox (transaction) mbxref);
    this.mbx=mbx;
    this.mbxref=mbxred;
  endfunction
  task run();
    repeat (count) begin
      for(int i=0;i<10;i++) begin
        mbx.put(tr.copy);
        mbxref.put(tr.copy);
        tr.display("Gen");
        @(sconext);
      end
    end
    ->done
  endtask
endclass
```

Driver:-

```
class driver ;
  transaction tr;
  mailbox (transaction) mbx;
  virtual Clk_Div_odd_if vif;
  function new(mailbox (transaction) mbx);
    this.mbx=mbx;
  endfunction
  task reset();
    vif.rst<=1'b1;
    repeat (5) @(posedge vif.clk);
    vif.rst<=1'b0;
    @(posedge vif.clk);
    $display("[DRV]:Reset done");
  endtask
  task run();
    forever begin
      mbx.get(tr);
      tr.display("DRV");
      @(posedge vif.clk);
    end
  endtask
endclass
```

Monitor:-

```
class monitor;
transaction tr;
mailbox (transaction) mbx;
virtual clk_div_by_odd vif;
function new(mailbox (transaction) mbx);
this.mbx=mbx;
endfunction
task run();
tr=new();
forever begin
@(posedge vif.clk);
@(posedge vif.clk);
tr.div3=vif.div3;
tr.div5=vif.div5;
tr.div9=vif.div9;
mbx.put(tr);
tr.display("MON");
end
endtask
endclass
```

Scoreboard:-

```
class scoreboard;
transaction tr;
mailbox (transaction) mbx;
mailbox (transaction) mbxref;
event sconext;
function new(mailbox (transaction) mbx,mailbox (transaction) mbxref);
this.mbx=mbx;
this.mbxref=mbxref;
endfunction
task run();
forever begin
mbx.get(tr);
mbx.get(tr);
tr.display("SCO");
if(tr.div3==tr.di2ref && tr.div3==tr.di2ref && tr.div3==tr.di2ref)
$display("Data matched");
else
$display("Data Not matched");
->sconext;
end
endtask
endclass
```

Environment:-

```
class environment;
  generator gen;
  scoreboard sco;
  monitor mon;
  driver drv;
  event next;
  mailbox #(transaction) gdmbx; ///gen - drv
  mailbox #(transaction) msmbx;  /// mon - sco
  mailbox #(transaction) mbxref; ///// gen -> sco
  virtual clk_Div_if vif;
  function new(virtual clk_Div_if vif);
    gdmbx=new();
    msmbx=new();
    mbxref=new();
    gen=new(gdmbx,mbxref);
    sco=new(msmbx,mbxref);
    this.vif=vif;
    drv.vif=this.vif;
    mon.vif=this.vif;
    gen.sconext=next;
    sco.sconext=next;
  endfunction
  task pre_test();
    drv.reset();
  endtask
  task test();
  fork
    gen.run();
    drv.run();
    mon.run();
    sco.run();
  join_any
  endtask
  task post_test();
    wait(gen.done.triggered);
    $finish();
  endtask
  task run();
    pre_test();
    test();
    post_test();
  endtask
endclass
```

TopModule:-

```
module topmodule;
  reg clk,rst;
  wire div3,div5,div9;
  Clk_Div_by_odd DUT(.rst(0),.clk(clk),.div3(div3),.div5(div5),.div9(div9));
  initial clk=0;
  always #5 clk=~clk;
  environment env;
  initial begin
    env=new(vif);
    env.gen.count(30);
    env.run();
  end
  initial begin
    $dumpfile("file.vcd");
    $dumpvars;
  end
endmodule
```