Experiment 5

Design of a verification environment for a 2-bit adder

AIM:

To design a complete verification environment for a clocked adder unit.

TOOLS USED: ModelSim, QuestaSim, EDA playground.

Questions:

Design a verification environment for a 2 bit adder that adds 2 numbers and updates the results on the rising edge of the clock. The environment should include monitor block along with the standard transactor, generator, driver, environment, test and any other additional blocks.

System Verilog Code with Comments (#):

Design.sv

```
/*
 Module : Adder (design.sv)
 19ECE347 - ELECTRONIC SYSTEM LEVEL DESIGN AND VERIFICATION ~ ASSIGNMENT 6
labsheet#5
 Roll no - CB.EN.U4ECE20101
 Change history: 01/12/23 - V1.0 - Initial working version created (owner:
Abinav-CB.EN.U4ECE20101)
*/


module adder (input clk,reset, valid,
              input [1:0] a, input [1:0] b,
              output [3:0] c);

  reg [3:0] c_temp;

  //reset
  always @(posedge reset)
    c_temp <= 0;
  //add operation
  always @(posedge clk)
    if(valid)
      c_temp <=  a+b;

  assign c = c_temp;

endmodule
```

Test bench top :

```
/*
 module tbtop part of the testbench for : Adder (design.sv)
 19ECE347 - ELECTRONIC SYSTEM LEVEL DESIGN AND VERIFICATION ~ ASSIGNMENT 6
labsheet#5
 Roll no - CB.EN.U4ECE20101
 Change history: 01/12/23 - V1.0 - Initial working version created (owner:
Abinav-CB.EN.U4ECE20101)
*/

`include "interface.sv"
`include "randtest.sv"

module tbtop;
  bit clk;
  bit reset;

  always #5 clk = ~clk;

  initial begin
    reset = 1;
    #5 reset =0;
  end

  intf i_intf(clk,reset);
  test t1(i_intf);

   adder DUT (
    .clk(i_intf.clk),
    .reset(i_intf.reset),
    .a(i_intf.a),
    .b(i_intf.b),
    .valid(i_intf.valid),
    .c(i_intf.c)
   );
  // I have added this so that the op waveform can be observed
    initial begin
    $dumpfile("dump.vcd"); $dumpvars;
  end
endmodule
```

Transaction. Sv

```
/*
 Class transaction.sv part of testbench design : Adder (design.sv)
 19ECE347 - ELECTRONIC SYSTEM LEVEL DESIGN AND VERIFICATION ~ ASSIGNMENT 6
labsheet#5
 Roll no - CB.EN.U4ECE20101
 Change history: 01/12/23 - V1.0 - Initial working version created (owner:
Abinav-CB.EN.U4ECE20101)

*/

class transaction;

  rand bit [1:0] a;
  rand bit [1:0] b;
       bit [3:0] c;
  function void display(string name);
    $display("-------------------------");
    $display("- %s ",name);
    $display("-------------------------");
    $display("- a = %0d, b = %0d",a,b);
    $display("- c = %0d",c);
    $display("-------------------------");
  endfunction
endclass
```

Driver.sv

```
/*
 Class driver part of the testbench for : Adder (design.sv)
 19ECE347 - ELECTRONIC SYSTEM LEVEL DESIGN AND VERIFICATION ~ ASSIGNMENT 6
labsheet#5
 Roll no - CB.EN.U4ECE20101
 Change history: 01/12/23 - V1.0 - Initial working version created (owner:
Abinav-CB.EN.U4ECE20101)
*/

class driver;

  int no_transactions; //count total transactions
  virtual intf vif;   // virtual interface

  mailbox gen2driv;
  function new(virtual intf vif, mailbox gen2driv); //constructor function
    this.vif = vif;
    this.gen2driv = gen2driv;
  endfunction

  task reset;
    wait(vif.reset);
    $display("Reset Started");
    vif.a<= 0;
    vif.b<= 0;
    vif.valid<= 0 ;
    wait(!vif.reset);
    $display("Reset Ended");
  endtask

  task main;
    forever begin
      transaction trans;
      gen2driv.get(trans);
      @(posedge vif.clk);
      vif.valid<=1;
      vif.a <= trans.a;
      vif.b <= trans.b;
      @(posedge vif.clk)
      vif.valid <= 0;
      trans.c   = vif.c;
      @(posedge vif.clk);
      trans.display("[ Driver ]");
      no_transactions++;
    end
  endtask

endclass
```

Honour Pledge: I affirm that I will not give or receive any unauthorized help on this lab, and that all work will be my own

Environment.Sv

```systemverilog
/*
 Class environment part of the testbench for : Adder (design.sv)
 19ECE347 - ELECTRONIC SYSTEM LEVEL DESIGN AND VERIFICATION ~ ASSIGNMENT 6
labsheet#5
 Roll no - CB.EN.U4ECE20101
 Change history: 01/12/23 - V1.0 - Initial working version created (owner:
Abinav-CB.EN.U4ECE20101)
*/

`include "transaction.sv"
`include "generator.sv"
`include "Driver.sv"
class environment;


  generator gen;
  driver    driv;
  mailbox gen2driv;
  virtual intf vif;

  //constructor
  function new(virtual intf vif);
    //get the interface from test
    this.vif = vif;
    gen2driv = new();
    gen  = new(gen2driv);
    driv = new(vif,gen2driv);
  endfunction

  //divide into 3 different tasks and call them all in one task to simplify
everything.

  task pre_test();
    driv.reset();
  endtask



  task test();
    fork
    gen.main();
    driv.main();
    join_any
```

Honour Pledge: I affirm that I will not give or receive any unauthorized help on this lab, and that all work will be my own

```
  endtask


  task post_test();
    wait(gen.ended.triggered);
    wait(gen.repeat_count == driv.no_transactions);
  endtask


  task run;
    pre_test();
    test();
    post_test();
    $finish;
  endtask

endclass
```

Interface.sv

```
/*
 interface part of the testbench for : Adder (design.sv)
 19ECE347 - ELECTRONIC SYSTEM LEVEL DESIGN AND VERIFICATION ~ ASSIGNMENT 6
labsheet#5
 Roll no - CB.EN.U4ECE20101
 Change history: 01/12/23 - V1.0 - Initial working version created (owner:
Abinav-CB.EN.U4ECE20101)
*/

interface intf(input logic clk,reset);


  logic valid;
  logic [1:0] a;
  logic [1:0] b;
  logic [3:0] c;

endinterface
```

Honour Pledge: I affirm that I will not give or receive any unauthorized help on this lab, and that all work will be my own

Generator.Sv

```
/*
 Class generator part of the testbench for : Adder (design.sv)
 19ECE347 - ELECTRONIC SYSTEM LEVEL DESIGN AND VERIFICATION ~ ASSIGNMENT 6
labsheet#5
 Roll no - CB.EN.U4ECE20101
 Change history: 01/12/23 - V1.0 - Initial working version created (owner:
Abinav-CB.EN.U4ECE20101)
*/

class generator;
  rand transaction trans;
  int repeat_count;
  mailbox gen2driv;
  event ended;

  function new(mailbox gen2driv);
    this.gen2driv = gen2driv;
  endfunction

  task main();
    repeat(repeat_count)begin
    trans = new();
      if( !trans.randomize() )
      trans.display("[ Generator ]");
      gen2driv.put(trans);
    end
    -> ended; //event trigger operation.
  endtask

endclass
```

Monitor and scoreboard:

```
`class monitor;

  //creating virtual interface handle
  virtual intf vif;

  //creating mailbox handle
  mailbox mon2scb;

  //constructor
  function new(virtual intf vif,mailbox mon2scb);
```

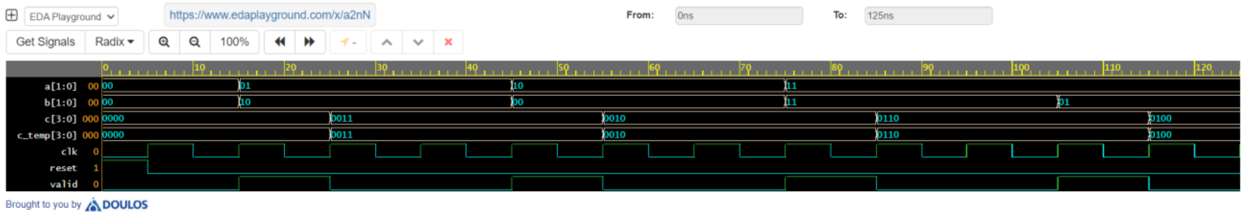
Honour Pledge: I affirm that I will not give or receive any unauthorized help on this lab, and that all work will be my own

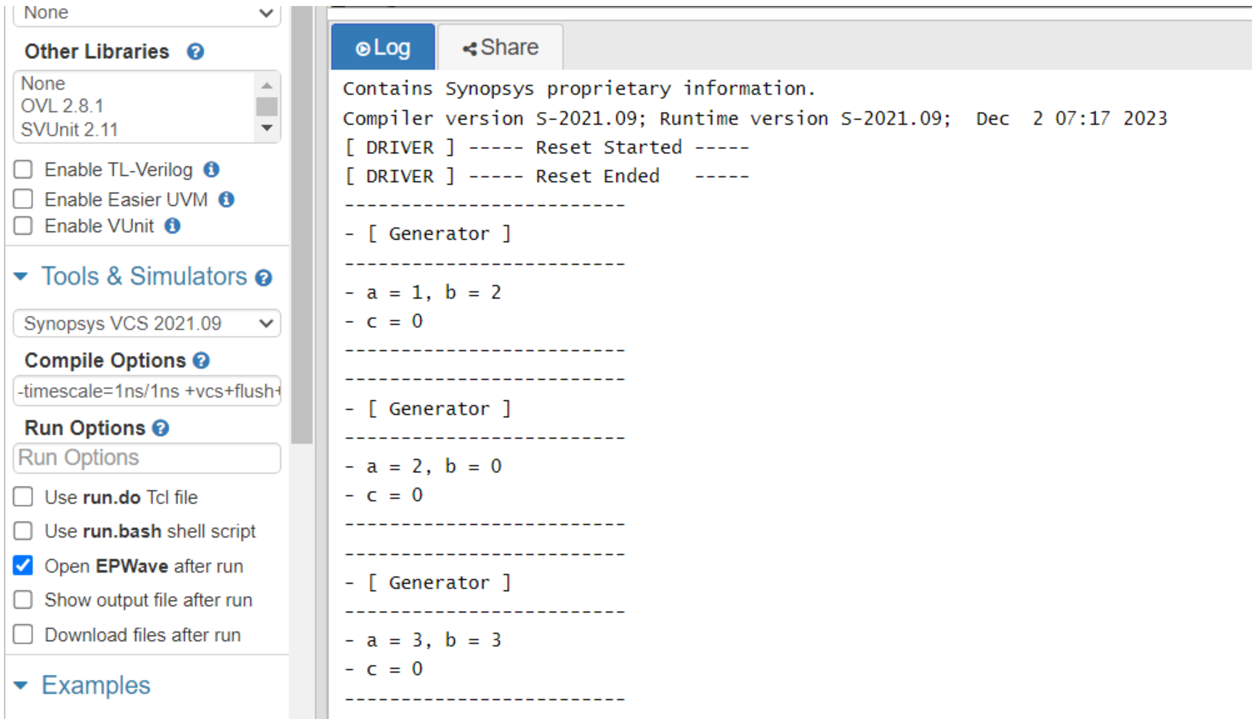
```
    //getting the interface
    this.vif = vif;
    //getting the mailbox handles from  environment
    this.mon2scb = mon2scb;
  endfunction

  //Samples the interface signal and send the sample packet to scoreboard
  task main;
    forever begin
      transaction trans;
      trans = new();
      @(posedge vif.clk);
      wait(vif.valid);
      trans.a   = vif.a;
      trans.b   = vif.b;
      @(posedge vif.clk);
      trans.c   = vif.c;
      @(posedge vif.clk);
      mon2scb.put(trans);
      trans.display("[ Monitor ]");
    end
  endtask

endclass
class scoreboard;

  //creating mailbox handle
  mailbox mon2scb;

  //used to count the number of transactions
  int no_transactions;

  //constructor
  function new(mailbox mon2scb);
    //getting the mailbox handles from  environment
    this.mon2scb = mon2scb;
  endfunction

  //Compares the Actual result with the expected result
  task main;
    transaction trans;
    forever begin
      mon2scb.get(trans);
        if((trans.a+trans.b) == trans.c)
          $display("Result is as Expected");
```

Honour Pledge: I affirm that I will not give or receive any unauthorized help on this lab, and that all work will be my own

```
        else
          $error("Wrong Result.\n\tExpeced: %0d Actual:
%0d",(trans.a+trans.b),trans.c);
        no_transactions++;
      trans.display("[ Scoreboard ]");
    end
  endtask

endclass
```

Results:





[ ] Enable TL-Verilog
[ ] Enable Easier UVM
[ ] Enable VUnit

Synopsys VCS 2021.09

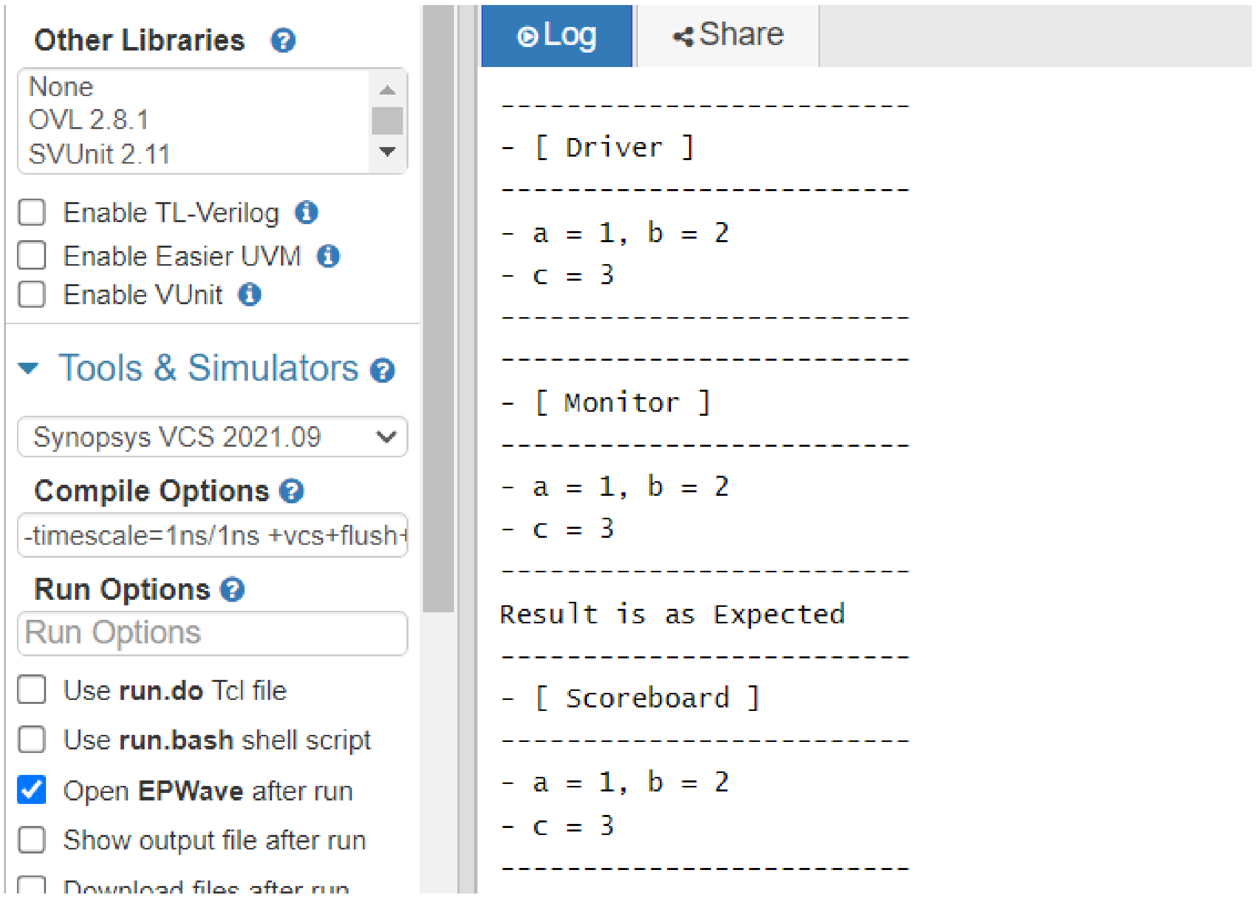
Compile Options: -timescale=1ns/1ns +vcs+flush+

Run Options: Run Options

[ ] Use run.do Tcl file
[ ] Use run.bash shell script
[✓] Open EPWave after run
[ ] Show output file after run
[ ] Download files after run

```
Contains Synopsys proprietary information.
Compiler version S-2021.09; Runtime version S-2021.09;  Dec  2 07:17 2023
[ DRIVER ] ----- Reset Started -----
[ DRIVER ] ----- Reset Ended   -----
 -------------------------
 - [ Generator ]
 -------------------------
 - a = 1, b = 2
 - c = 0
 -------------------------
 -------------------------
 - [ Generator ]
 -------------------------
 - a = 2, b = 0
 - c = 0
 -------------------------
 -------------------------
 - [ Generator ]
 -------------------------
 - a = 3, b = 3
 - c = 0
 -------------------------
```

Honour Pledge: I affirm that I will not give or receive any unauthorized help on this lab, and that all work will be my own



Synopsys VCS 2021.09

Compile Options: -timescale=1ns/1ns +vcs+flush

- [ ] Use run.do Tcl file
- [ ] Use run.bash shell script
- [x] Open EPWave after run
- [ ] Show output file after run
- [ ] Download files after run

```
-------------------------
- [ Driver ]
-------------------------
- a = 1, b = 2
- c = 3
-------------------------
-------------------------
- [ Monitor ]
-------------------------
- a = 1, b = 2
- c = 3
-------------------------
Result is as Expected
-------------------------
- [ Scoreboard ]
-------------------------
- a = 1, b = 2
- c = 3
-------------------------
```

Inference:

We have designed a verification environment for a 2-bit adder which has a monitor and scoreboard to verify the outputs along with the reset of the normal testbench blocks.

Code link : https://www.edaplayground.com/x/a2nN

Honour Pledge: I affirm that I will not give or receive any unauthorized help on this lab, and that all work will be my own