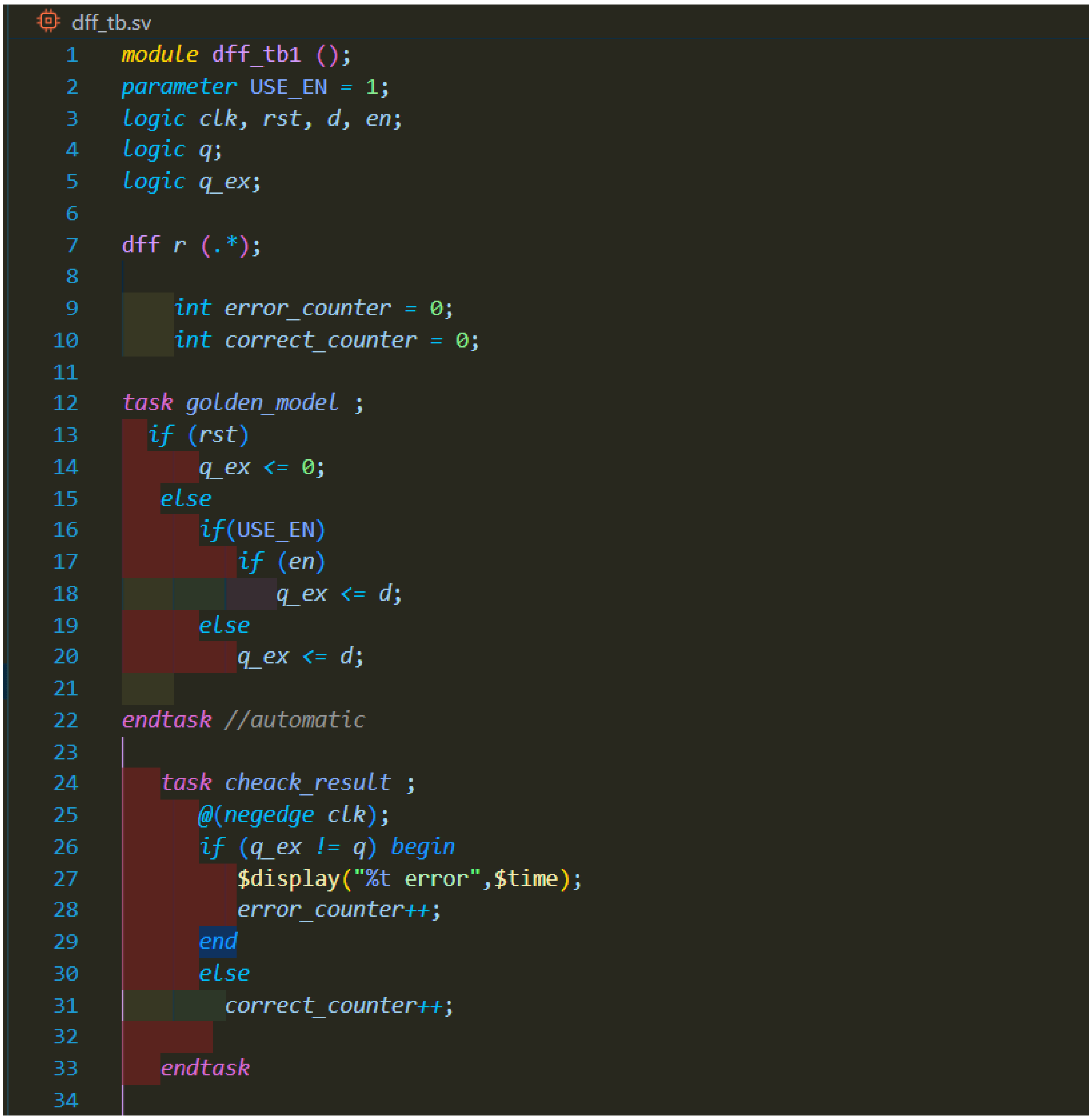
Assignment 1 extra

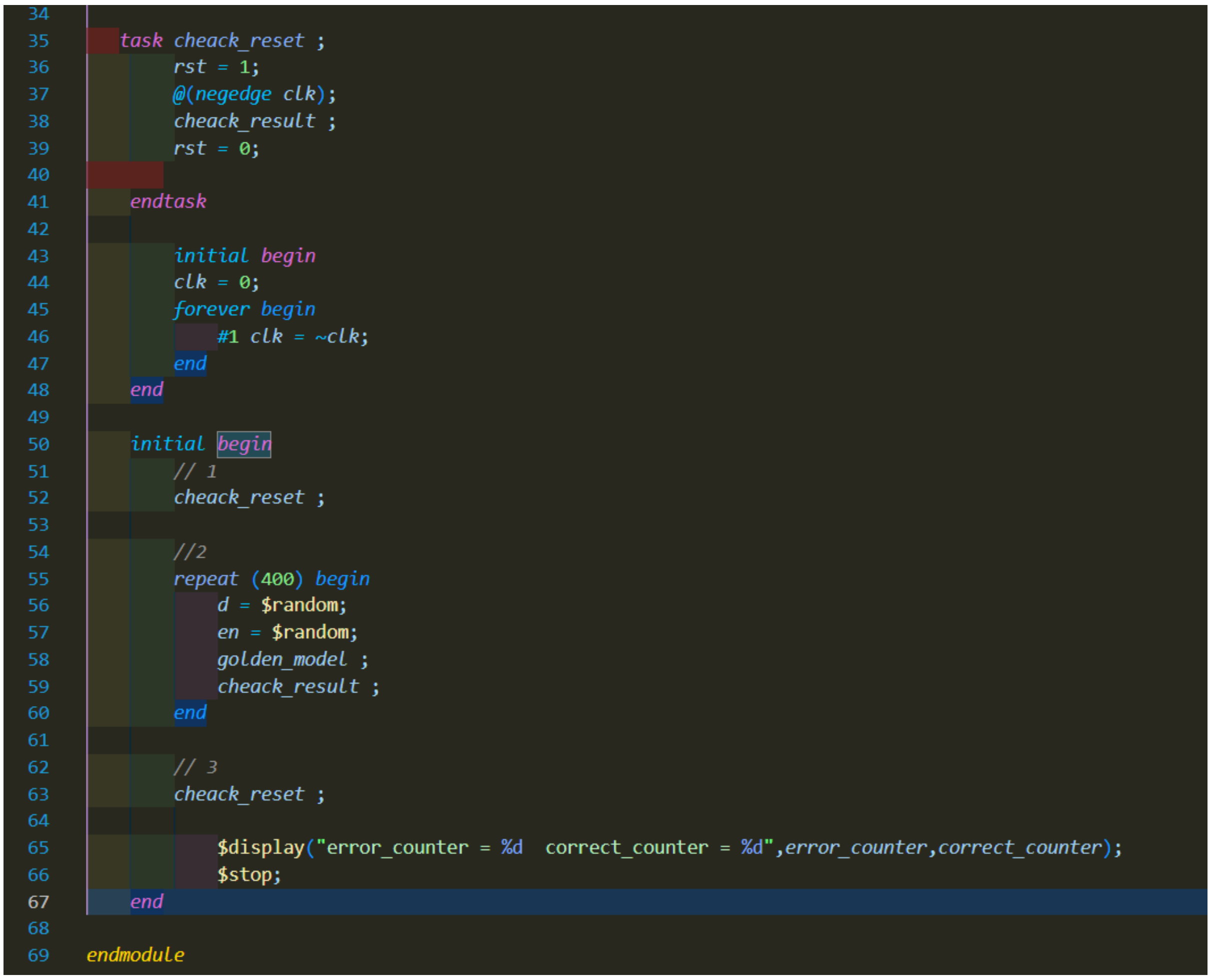
Q1

CODE TB 1 :



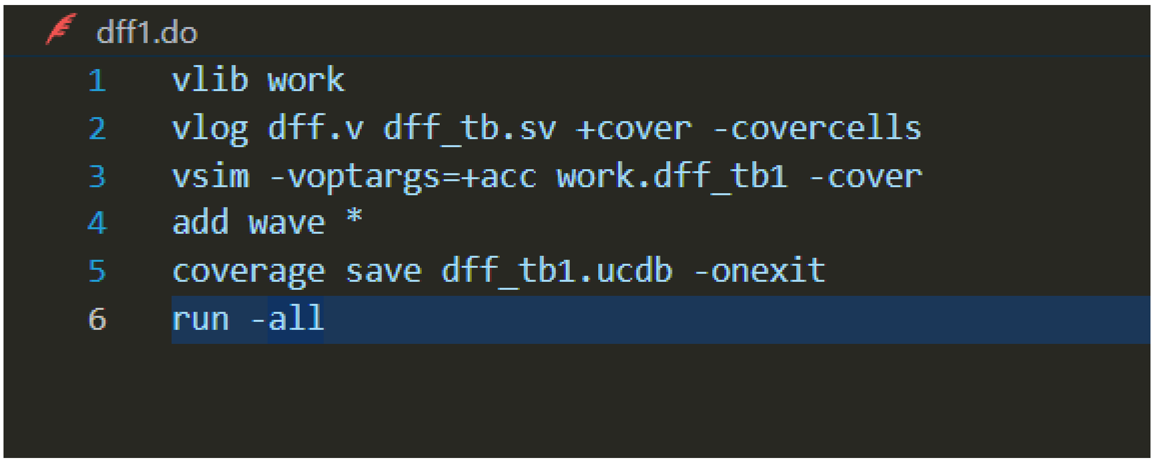
dff_tb.sv

```
module dff_tb1 ();
parameter USE_EN = 1;
logic clk, rst, d, en;
logic q;
logic q_ex;

dff r (.*);

    int error_counter = 0;
    int correct_counter = 0;

task golden_model ;
  if (rst)
        q_ex <= 0;
   else
        if(USE_EN)
           if (en)
              q_ex <= d;
        else
           q_ex <= d;

endtask //automatic

   task cheack_result ;
        @(negedge clk);
        if (q_ex != q) begin
           $display("%t error",$time);
           error_counter++;
        end
        else
          correct_counter++;

   endtask

```



```
task cheack_reset ;
    rst = 1;
    @(negedge clk);
    cheack_result ;
    rst = 0;

  endtask

    initial begin
    clk = 0;
    forever begin
        #1 clk = ~clk;
    end
  end

  initial begin
    // 1
    cheack_reset ;

    //2
    repeat (400) begin
        d = $random;
        en = $random;
        golden_model ;
        cheack_result ;
    end

    // 3
    cheack_reset ;

        $display("error_counter = %d  correct_counter = %d",error_counter,correct_counter);
        $stop;
  end

endmodule
```

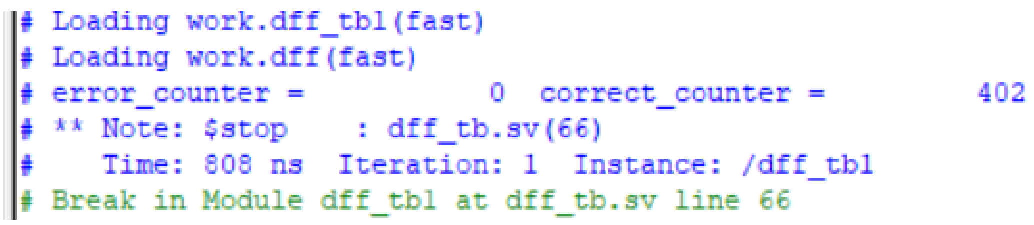
DO FILE 1 :



dff1.do

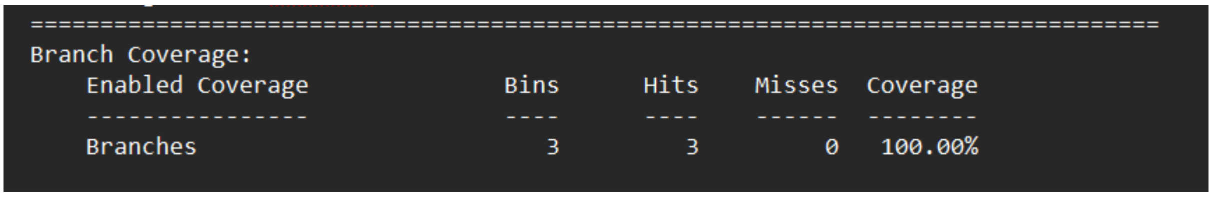
```
vlib work
vlog dff.v dff_tb.sv +cover -covercells
vsim -voptargs=+acc work.dff_tb1 -cover
add wave *
coverage save dff_tb1.ucdb -onexit
run -all
```

DISPLAY 1 :

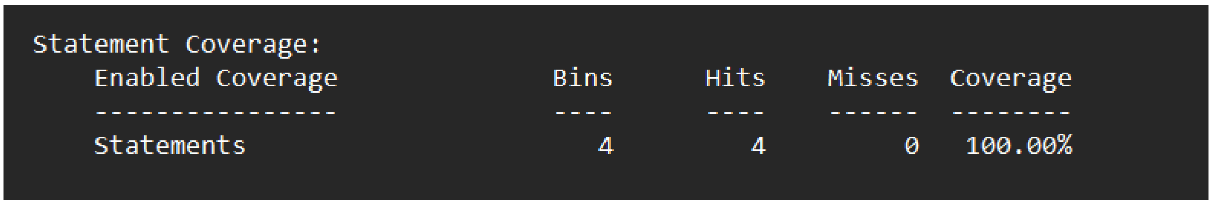


```
# Loading work.dff_tbl(fast)
# Loading work.dff(fast)
# error_counter =           0  correct_counter =         402
# ** Note: $stop    : dff_tb.sv(66)
#    Time: 808 ns  Iteration: 1  Instance: /dff_tbl
# Break in Module dff_tbl at dff_tb.sv line 66
```

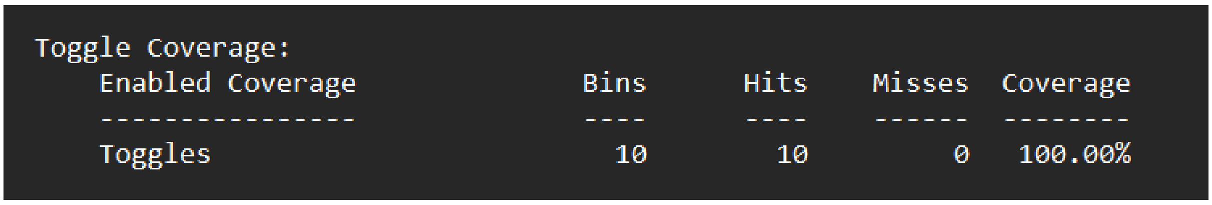
COVERAGE 1 :



```
====================================================================================
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         3         3         0   100.00%
```

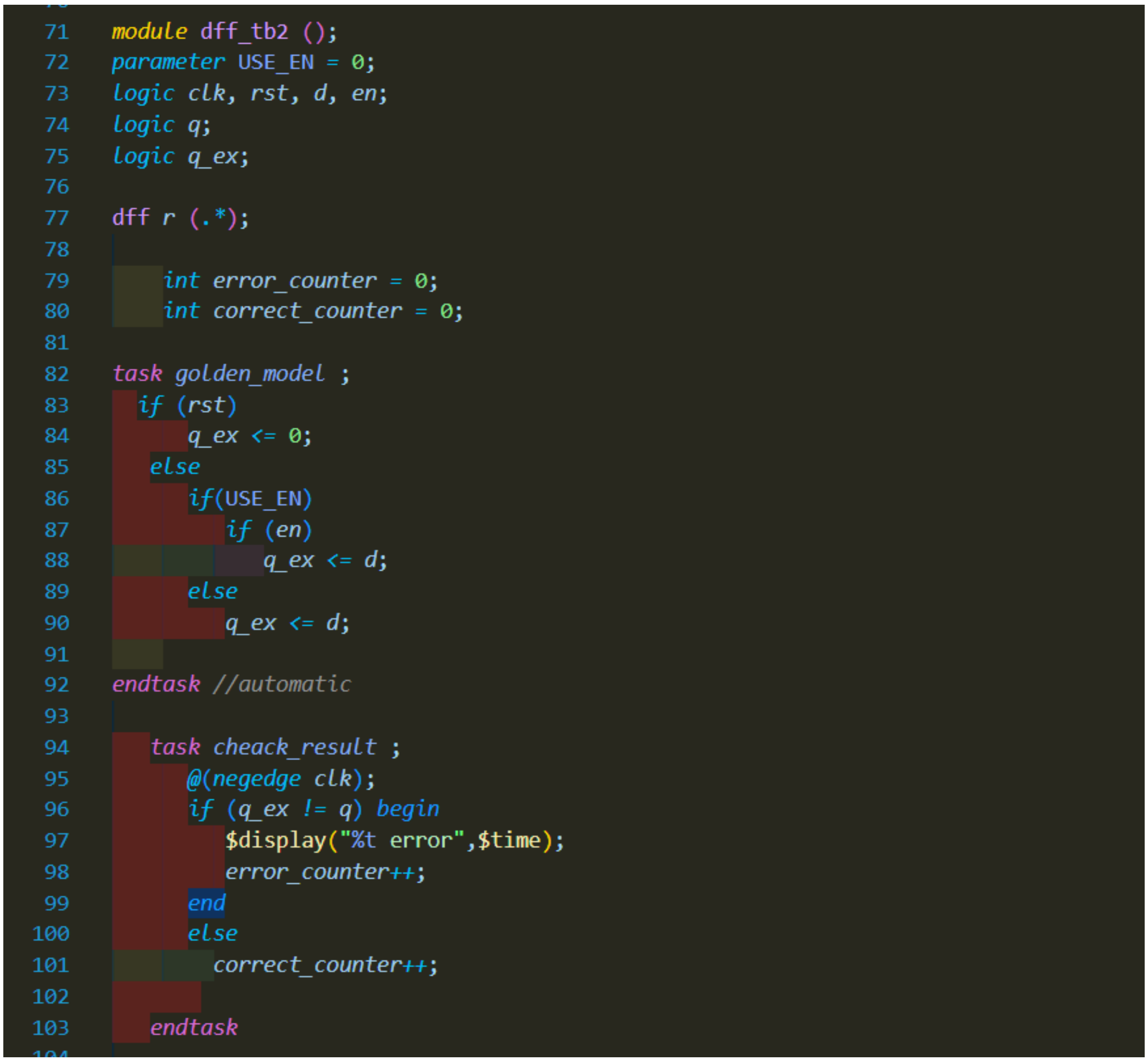


```
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                       4         4         0   100.00%
```

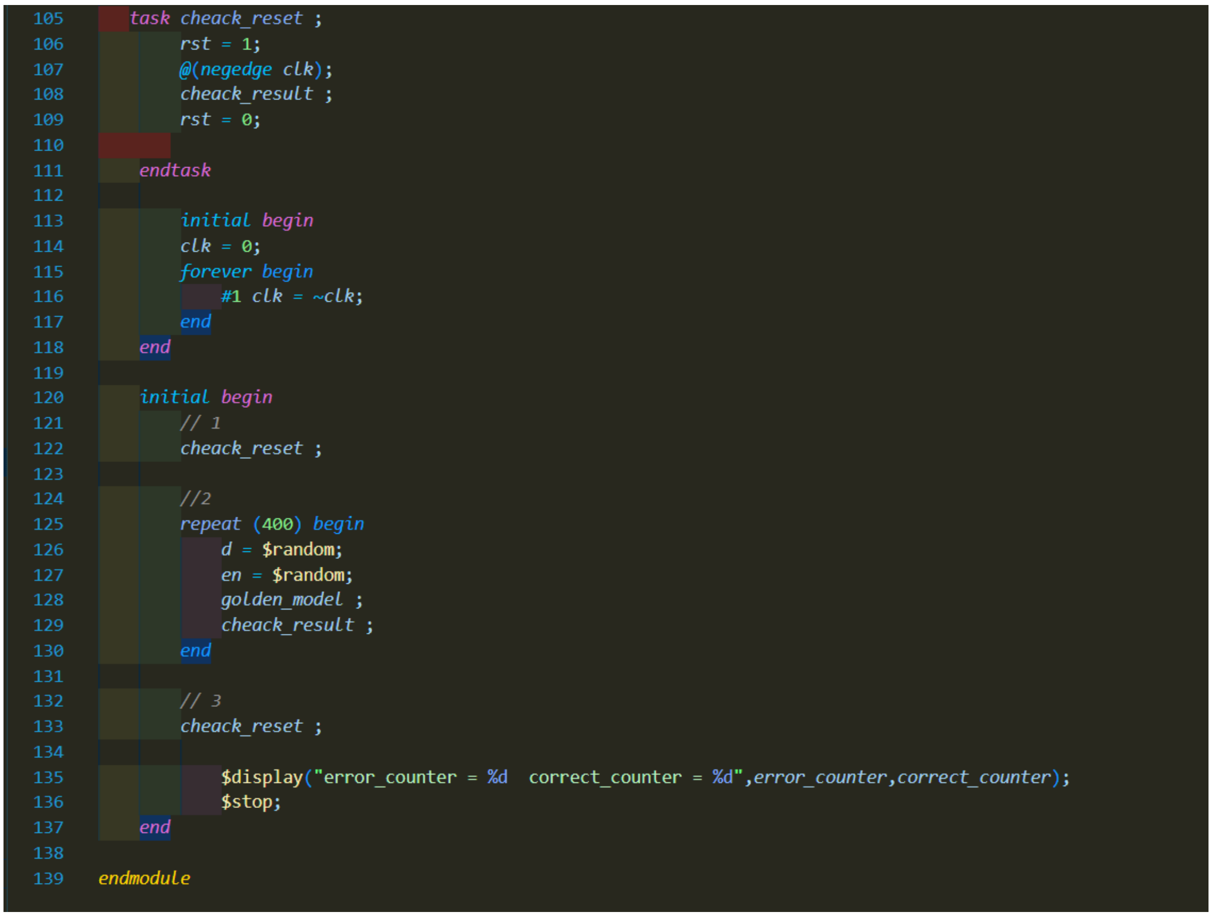


```
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         10        10         0   100.00%
```

CODE TB 2 :

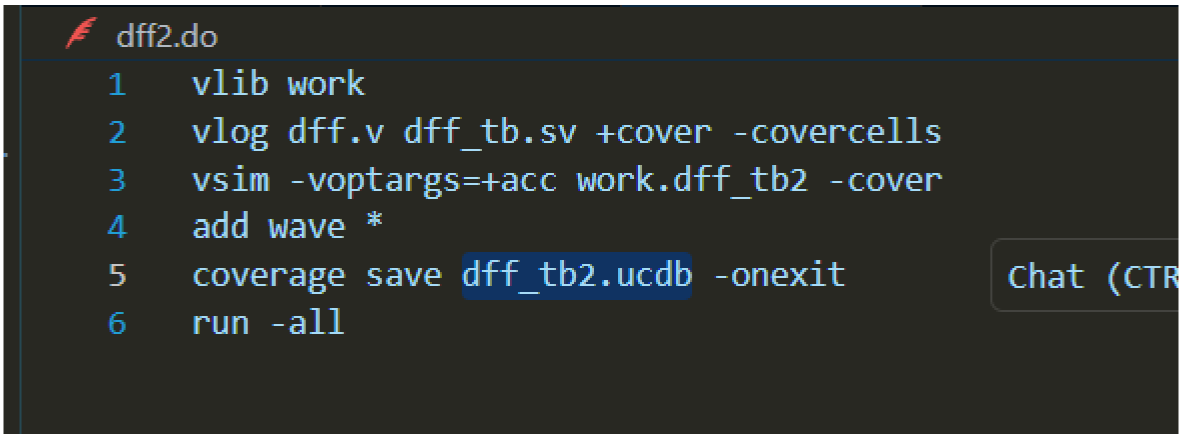


```
module dff_tb2 ();
parameter USE_EN = 0;
logic clk, rst, d, en;
logic q;
logic q_ex;

dff r (.*);

    int error_counter = 0;
    int correct_counter = 0;

task golden_model ;
  if (rst)
        q_ex <= 0;
   else
        if(USE_EN)
           if (en)
              q_ex <= d;
        else
           q_ex <= d;

endtask //automatic

   task cheack_result ;
        @(negedge clk);
        if (q_ex != q) begin
           $display("%t error",$time);
           error_counter++;
        end
        else
          correct_counter++;

   endtask
```



```
task cheack_reset ;
    rst = 1;
    @(negedge clk);
    cheack_result ;
    rst = 0;

 endtask

    initial begin
    clk = 0;
    forever begin
        #1 clk = ~clk;
    end
 end

 initial begin
    // 1
    cheack_reset ;

    //2
    repeat (400) begin
        d = $random;
        en = $random;
        golden_model ;
        cheack_result ;
    end

    // 3
    cheack_reset ;

        $display("error_counter = %d  correct_counter = %d",error_counter,correct_counter);
        $stop;
 end

endmodule
```

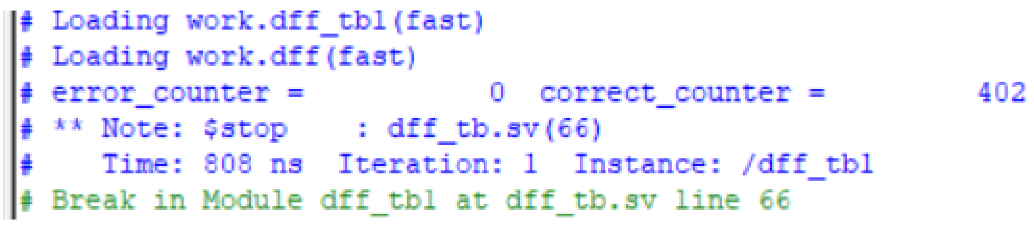
DO FILE 2 :



dff2.do

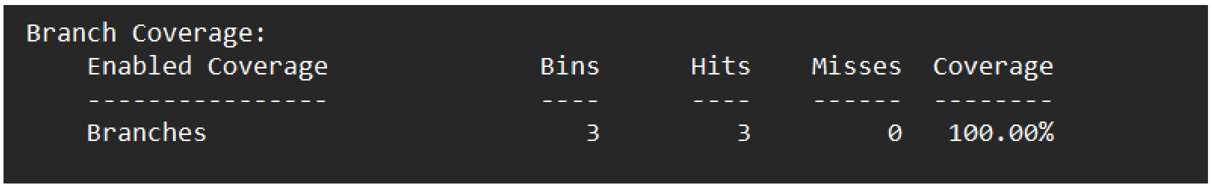
```
vlib work
vlog dff.v dff_tb.sv +cover -covercells
vsim -voptargs=+acc work.dff_tb2 -cover
add wave *
coverage save dff_tb2.ucdb -onexit
run -all
```

DISPLAY 2 :

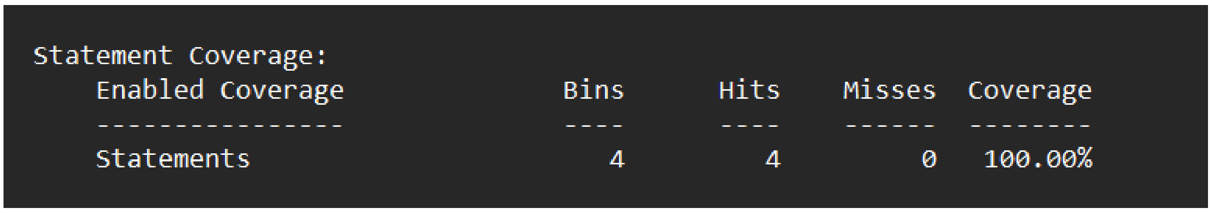


```
# Loading work.dff_tbl(fast)
# Loading work.dff(fast)
# error_counter =           0  correct_counter =         402
# ** Note: $stop    : dff_tb.sv(66)
#    Time: 808 ns  Iteration: 1  Instance: /dff_tbl
# Break in Module dff_tbl at dff_tb.sv line 66
```

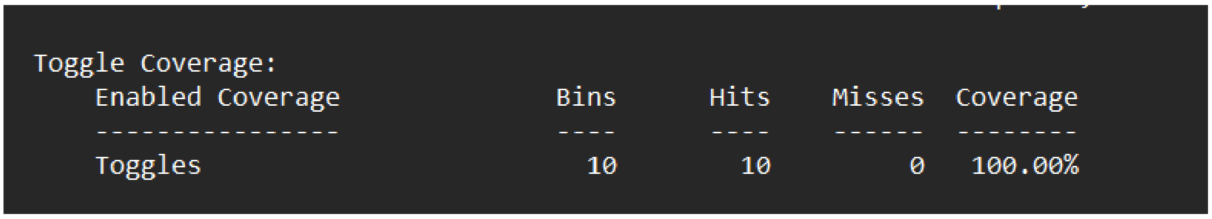
COVERAGE 2 :



```
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         3         3         0   100.00%
```

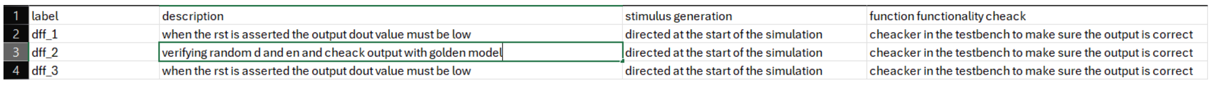


```
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                       4         4         0   100.00%
```



```
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         10        10         0   100.00%
```

VERIFICATION PLAN :



1	label	description	stimulus generation	function functionality cheack
2	dff_1	when the rst is asserted the output dout value must be low	directed at the start of the simulation	cheacker in the testbench to make sure the output is correct
3	dff_2	verifying random d and en and cheack output with golden model	directed at the start of the simulation	cheacker in the testbench to make sure the output is correct
4	dff_3	when the rst is asserted the output dout value must be low	directed at the start of the simulation	cheacker in the testbench to make sure the output is correct