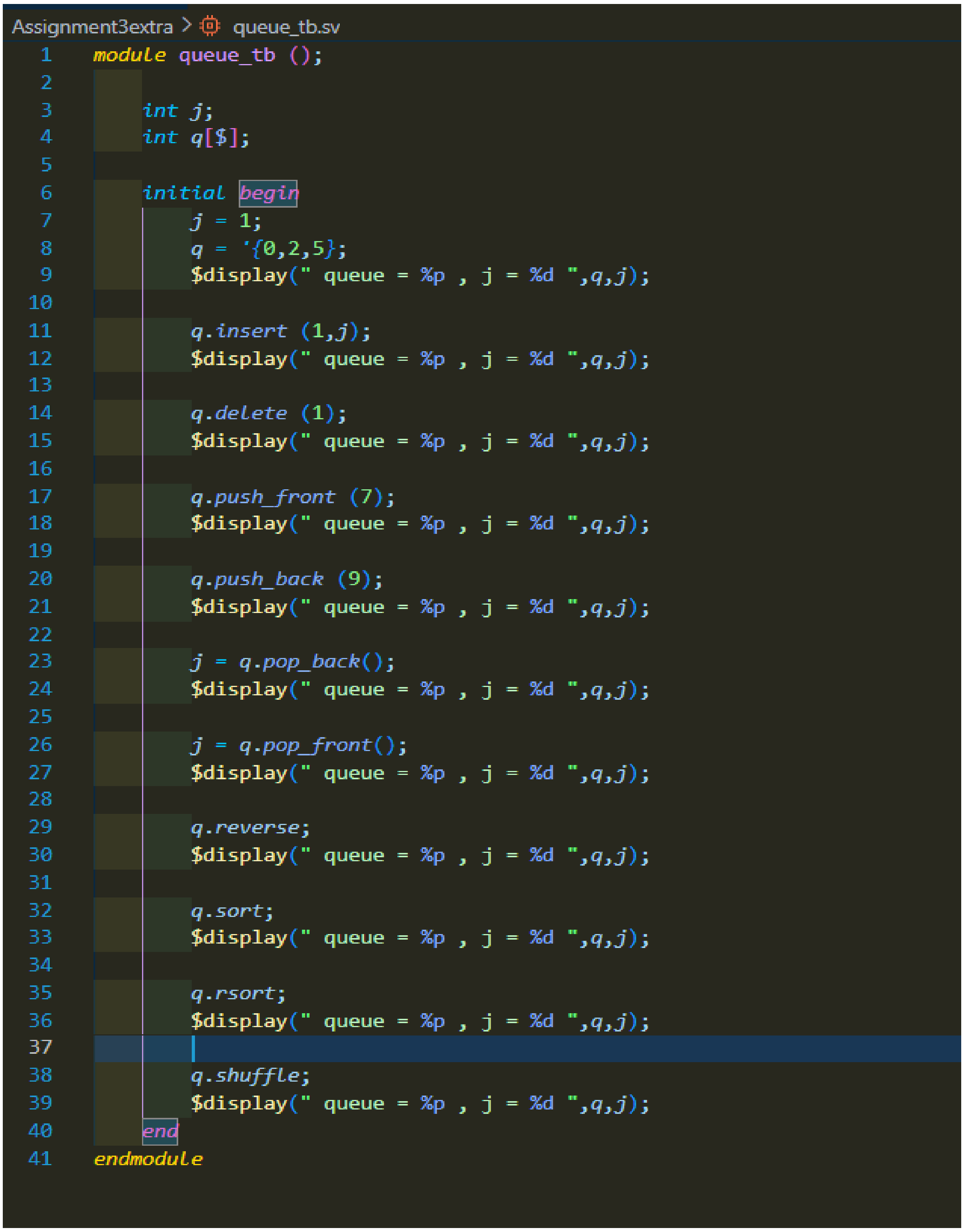
Assignment extra 3

Q1

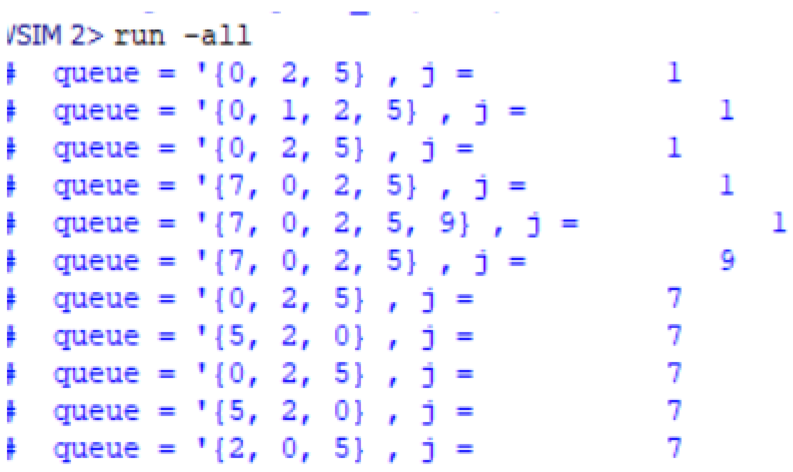
CODE :



Assignment3extra > queue_tb.sv

```
module queue_tb ();

    int j;
    int q[$];

    initial begin
        j = 1;
        q = '{0,2,5};
        $display(" queue = %p , j = %d ",q,j);

        q.insert (1,j);
        $display(" queue = %p , j = %d ",q,j);

        q.delete (1);
        $display(" queue = %p , j = %d ",q,j);

        q.push_front (7);
        $display(" queue = %p , j = %d ",q,j);

        q.push_back (9);
        $display(" queue = %p , j = %d ",q,j);

        j = q.pop_back();
        $display(" queue = %p , j = %d ",q,j);

        j = q.pop_front();
        $display(" queue = %p , j = %d ",q,j);

        q.reverse;
        $display(" queue = %p , j = %d ",q,j);

        q.sort;
        $display(" queue = %p , j = %d ",q,j);

        q.rsort;
        $display(" queue = %p , j = %d ",q,j);

        q.shuffle;
        $display(" queue = %p , j = %d ",q,j);
    end
endmodule
```

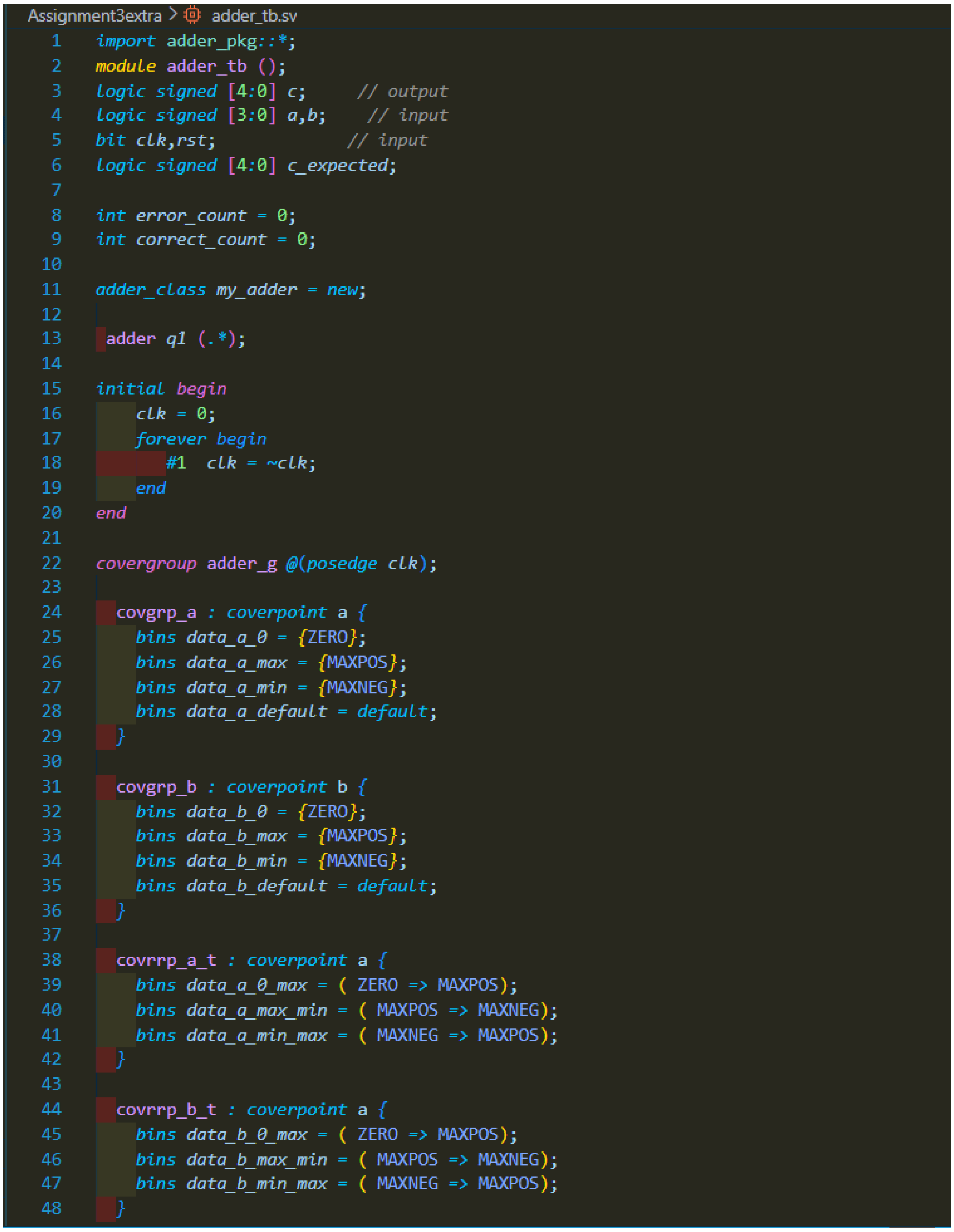
DISPLAY :



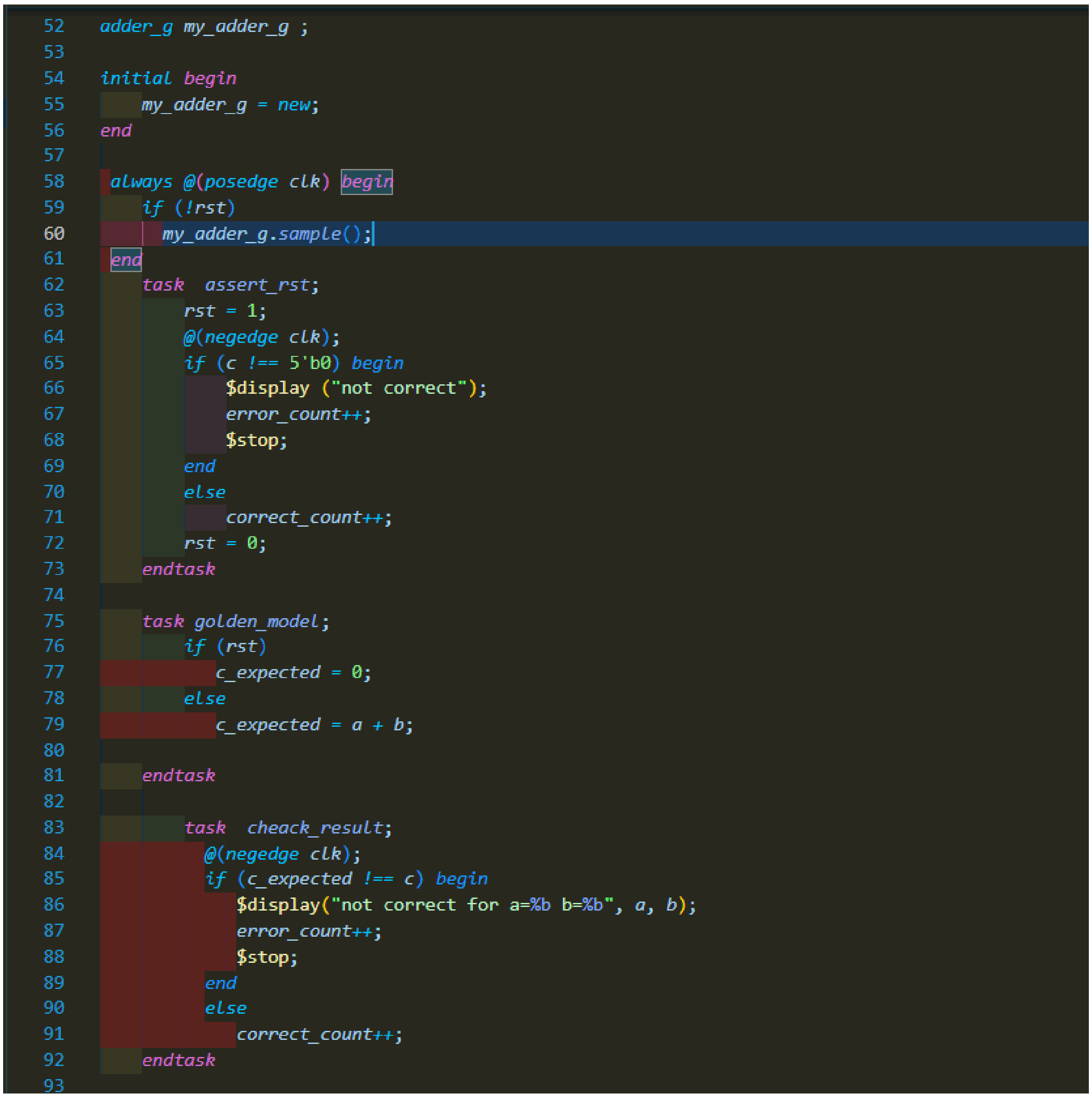
```
VSIM 2> run -all
# queue = '{0, 2, 5} , j =           1
# queue = '{0, 1, 2, 5} , j =           1
# queue = '{0, 2, 5} , j =           1
# queue = '{7, 0, 2, 5} , j =           1
# queue = '{7, 0, 2, 5, 9} , j =           1
# queue = '{7, 0, 2, 5} , j =           9
# queue = '{0, 2, 5} , j =           7
# queue = '{5, 2, 0} , j =           7
# queue = '{0, 2, 5} , j =           7
# queue = '{5, 2, 0} , j =           7
# queue = '{2, 0, 5} , j =           7
```

Q2

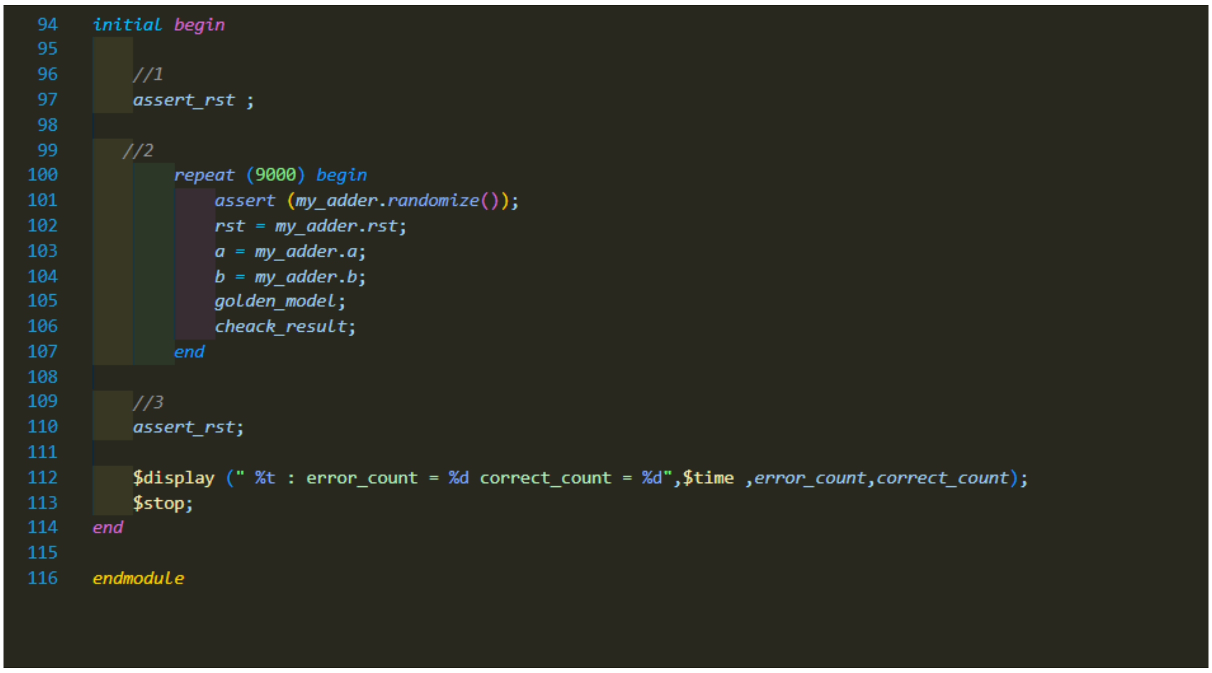
CODE TB :


Assignment3extra > adder_tb.sv

```
import adder_pkg::*;
module adder_tb ();
logic signed [4:0] c;     // output
logic signed [3:0] a,b;    // input
bit clk,rst;              // input
logic signed [4:0] c_expected;

int error_count = 0;
int correct_count = 0;

adder_class my_adder = new;

 adder q1 (.*);

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

covergroup adder_g @(posedge clk);

  covgrp_a : coverpoint a {
    bins data_a_0 = {ZERO};
    bins data_a_max = {MAXPOS};
    bins data_a_min = {MAXNEG};
    bins data_a_default = default;
  }

  covgrp_b : coverpoint b {
    bins data_b_0 = {ZERO};
    bins data_b_max = {MAXPOS};
    bins data_b_min = {MAXNEG};
    bins data_b_default = default;
  }

  covrrp_a_t : coverpoint a {
    bins data_a_0_max = ( ZERO => MAXPOS);
    bins data_a_max_min = ( MAXPOS => MAXNEG);
    bins data_a_min_max = ( MAXNEG => MAXPOS);
  }

  covrrp_b_t : coverpoint a {
    bins data_b_0_max = ( ZERO => MAXPOS);
    bins data_b_max_min = ( MAXPOS => MAXNEG);
    bins data_b_min_max = ( MAXNEG => MAXPOS);
  }
```

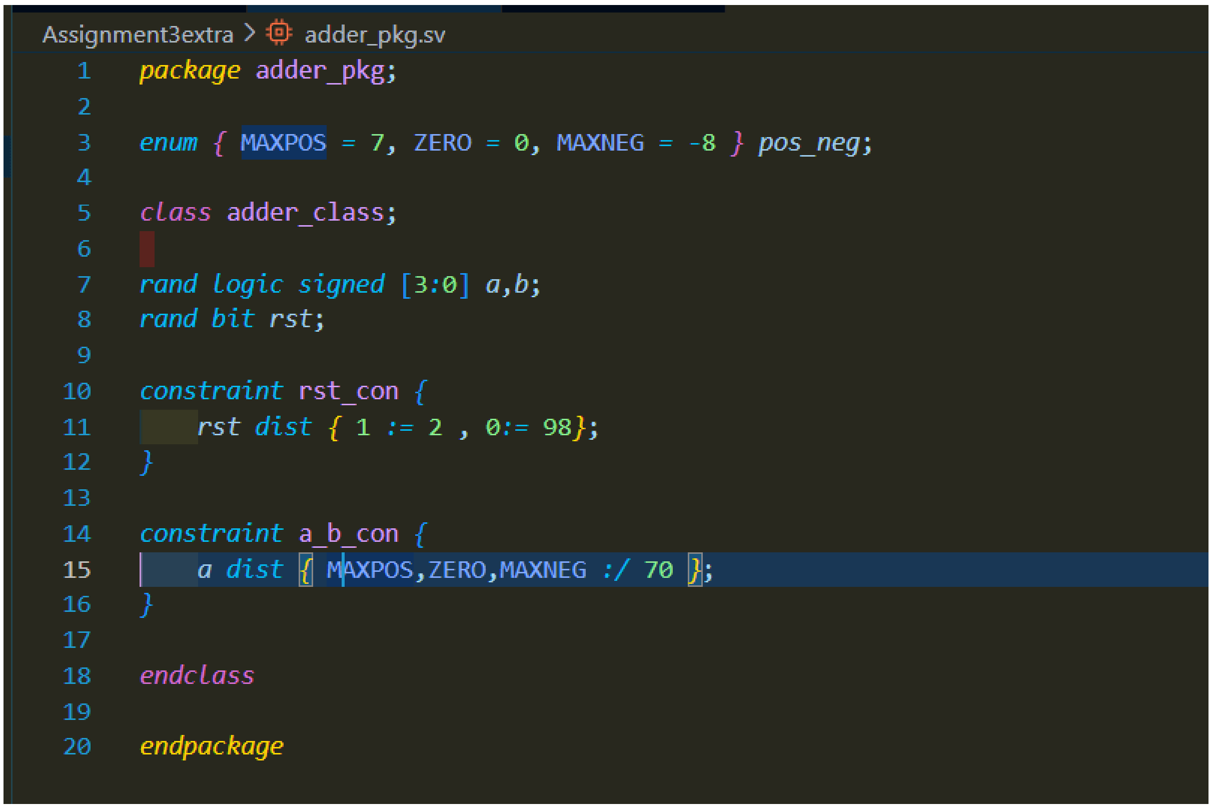


```
adder_g my_adder_g ;

initial begin
    my_adder_g = new;
end

always @(posedge clk) begin
    if (!rst)
      my_adder_g.sample();
end
    task  assert_rst;
        rst = 1;
        @(negedge clk);
        if (c !== 5'b0) begin
            $display ("not correct");
            error_count++;
            $stop;
        end
        else
            correct_count++;
        rst = 0;
    endtask

    task golden_model;
        if (rst)
           c_expected = 0;
        else
           c_expected = a + b;

    endtask

        task  cheack_result;
          @(negedge clk);
          if (c_expected !== c) begin
             $display("not correct for a=%b b=%b", a, b);
             error_count++;
             $stop;
          end
          else
             correct_count++;
    endtask

```



```
initial begin

    //1
    assert_rst ;

   //2
        repeat (9000) begin
            assert (my_adder.randomize());
            rst = my_adder.rst;
            a = my_adder.a;
            b = my_adder.b;
            golden_model;
            cheack_result;
        end

    //3
    assert_rst;

    $display (" %t : error_count = %d correct_count = %d",$time ,error_count,correct_count);
    $stop;
end

endmodule
```

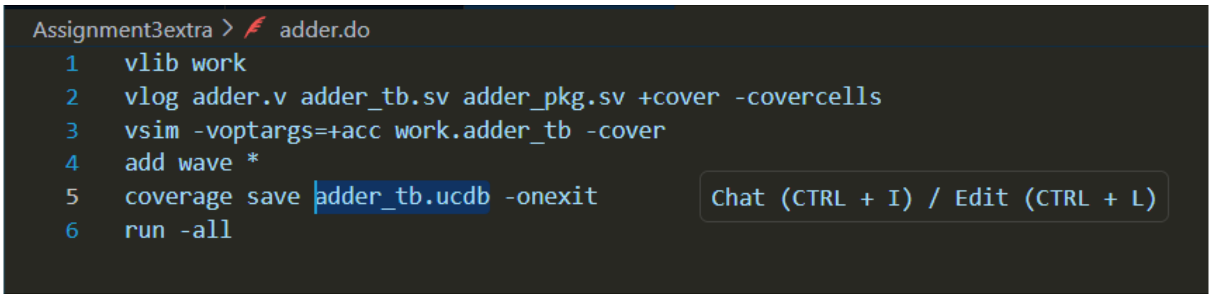
CODE PKG :



Assignment3extra > adder_pkg.sv

```
package adder_pkg;

enum { MAXPOS = 7, ZERO = 0, MAXNEG = -8 } pos_neg;

class adder_class;

rand logic signed [3:0] a,b;
rand bit rst;

constraint rst_con {
    rst dist { 1 := 2 , 0:= 98};
}

constraint a_b_con {
    a dist { MAXPOS,ZERO,MAXNEG :/ 70 };
}

endclass

endpackage
```

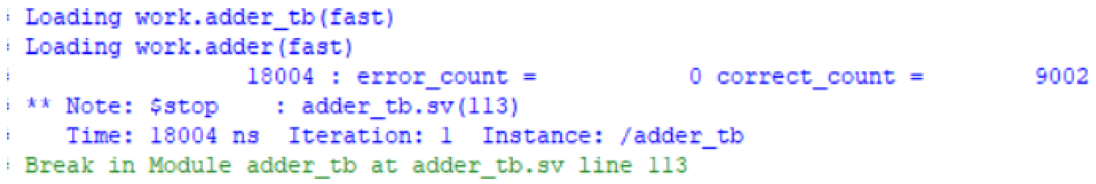
DO FILE :



```
Assignment3extra > adder.do
vlib work
vlog adder.v adder_tb.sv adder_pkg.sv +cover -covercells
vsim -voptargs=+acc work.adder_tb -cover
add wave *
coverage save adder_tb.ucdb -onexit
run -all
```

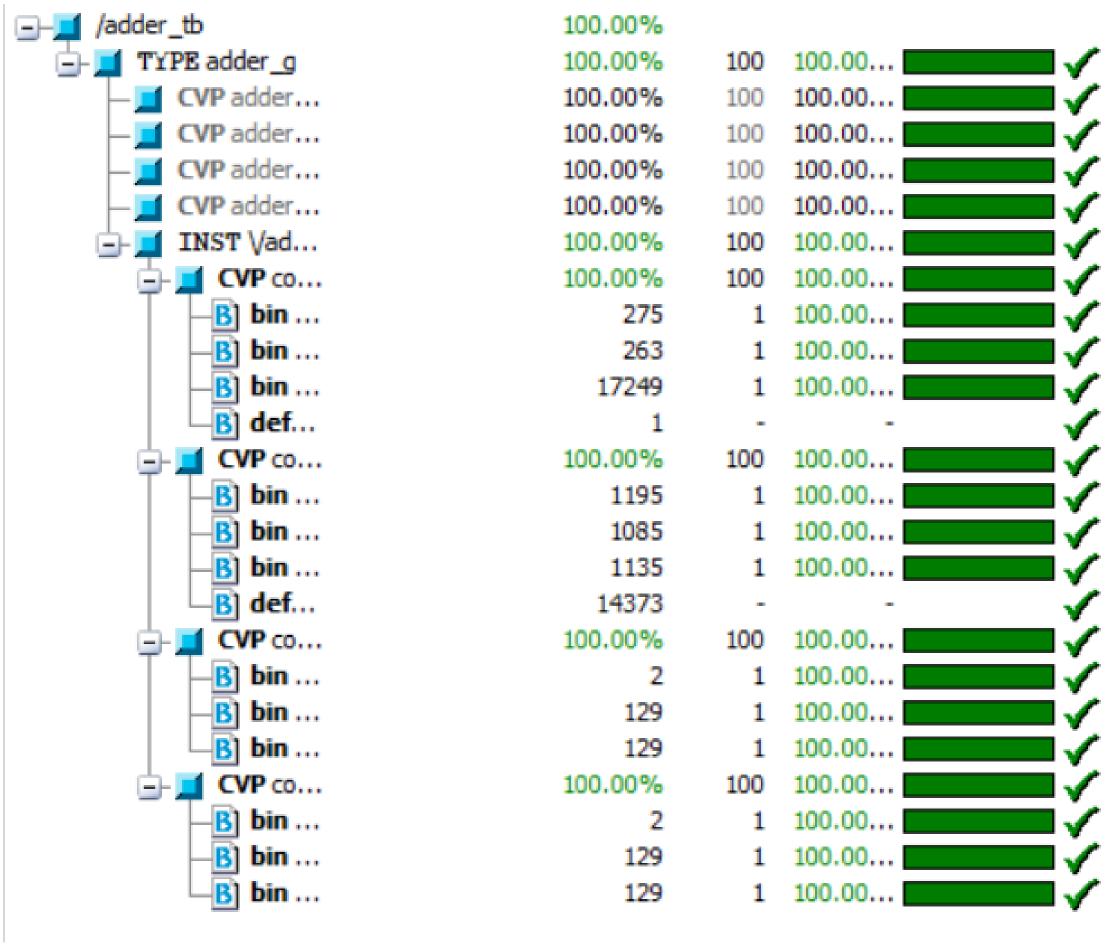
Chat (CTRL + I) / Edit (CTRL + L)

DISPLAY :



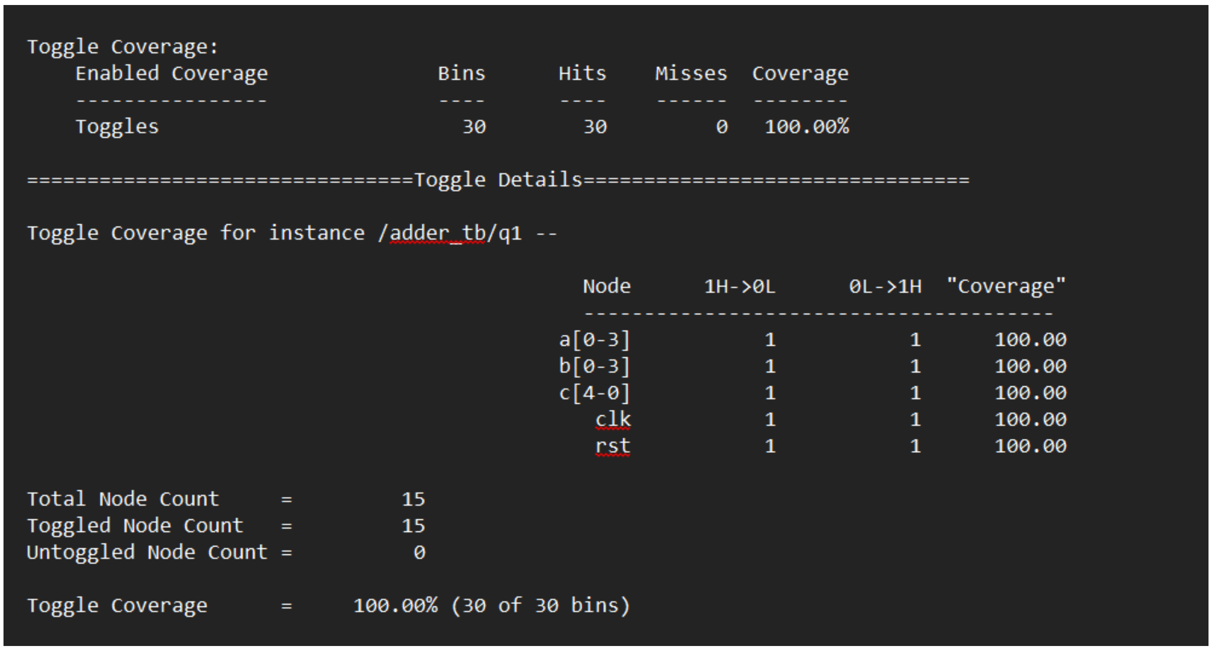
```
# Loading work.adder_tb(fast)
# Loading work.adder(fast)
#                 18004 : error_count =           0 correct_count =        9002
# ** Note: $stop    : adder_tb.sv(113)
#    Time: 18004 ns  Iteration: 1  Instance: /adder_tb
# Break in Module adder_tb at adder_tb.sv line 113
```

COVERGROUP :

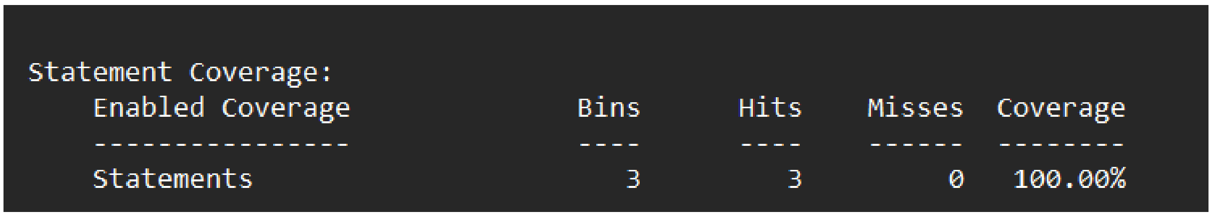


```
/adder_tb            100.00%
  TYPE adder_g       100.00%   100  100.00...  🟩 ✓
    CVP adder...     100.00%   100  100.00...  🟩 ✓
    CVP adder...     100.00%   100  100.00...  🟩 ✓
    CVP adder...     100.00%   100  100.00...  🟩 ✓
    CVP adder...     100.00%   100  100.00...  🟩 ✓
    INST \/ad...     100.00%   100  100.00...  🟩 ✓
      CVP co...      100.00%   100  100.00...  🟩 ✓
        bin ...          275     1  100.00...  🟩 ✓
        bin ...          263     1  100.00...  🟩 ✓
        bin ...        17249     1  100.00...  🟩 ✓
        def...             1     -          -     ✓
      CVP co...      100.00%   100  100.00...  🟩 ✓
        bin ...         1195     1  100.00...  🟩 ✓
        bin ...         1085     1  100.00...  🟩 ✓
        bin ...         1135     1  100.00...  🟩 ✓
        def...         14373     -          -     ✓
      CVP co...      100.00%   100  100.00...  🟩 ✓
        bin ...            2     1  100.00...  🟩 ✓
        bin ...          129     1  100.00...  🟩 ✓
        bin ...          129     1  100.00...  🟩 ✓
      CVP co...      100.00%   100  100.00...  🟩 ✓
        bin ...            2     1  100.00...  🟩 ✓
        bin ...          129     1  100.00...  🟩 ✓
        bin ...          129     1  100.00...  🟩 ✓
```

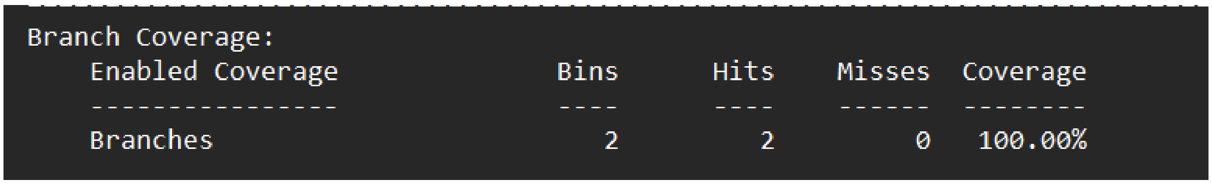
CODE COVERAGE :



```
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         30        30         0   100.00%

================================Toggle Details================================

Toggle Coverage for instance /adder_tb/q1 --

                                              Node      1H->0L      0L->1H  "Coverage"
                                              ---------------------------------------
                                            a[0-3]           1           1      100.00
                                            b[0-3]           1           1      100.00
                                            c[4-0]           1           1      100.00
                                               clk           1           1      100.00
                                               rst           1           1      100.00

Total Node Count     =         15
Toggled Node Count   =         15
Untoggled Node Count =          0

Toggle Coverage      =     100.00% (30 of 30 bins)
```



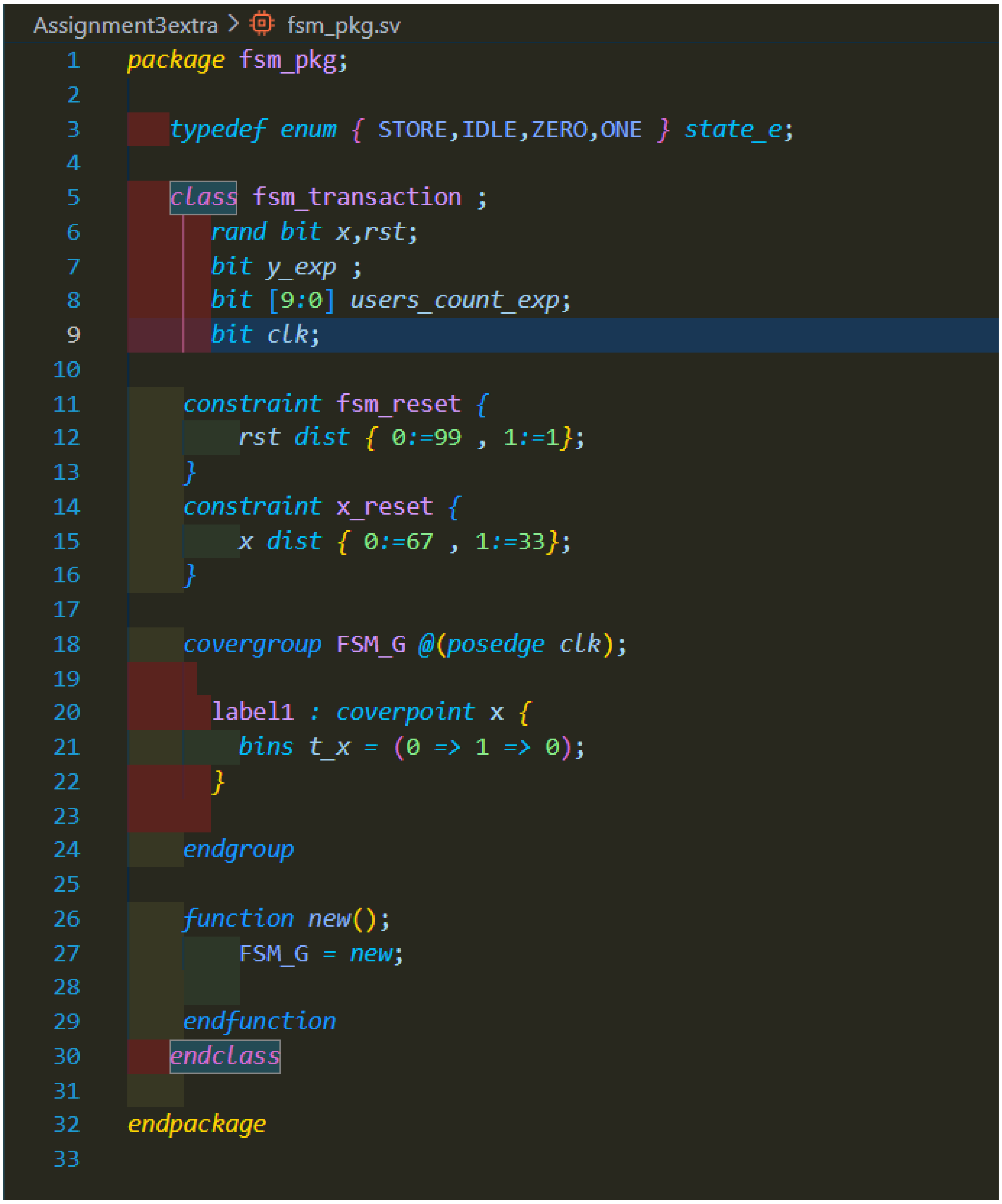
```
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                       3         3         0   100.00%
```



```
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         2         2         0   100.00%
```

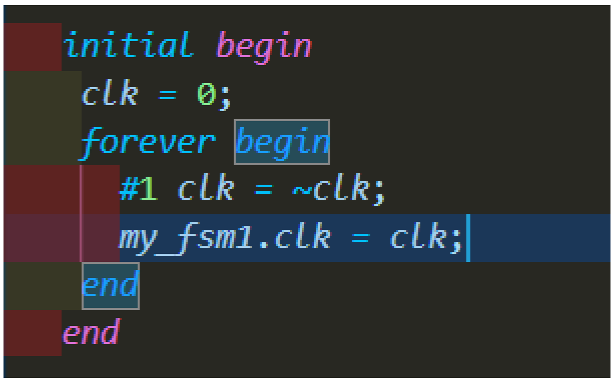
Q3

CODE PKG :



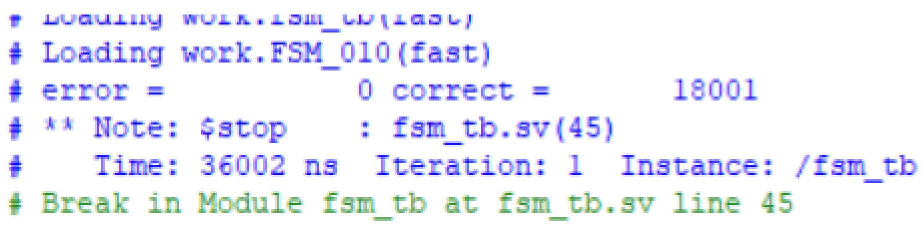
Assignment3extra > fsm_pkg.sv

```
package fsm_pkg;

typedef enum { STORE,IDLE,ZERO,ONE } state_e;

class fsm_transaction ;
    rand bit x,rst;
    bit y_exp ;
    bit [9:0] users_count_exp;
    bit clk;

  constraint fsm_reset {
      rst dist { 0:=99 , 1:=1};
  }
  constraint x_reset {
      x dist { 0:=67 , 1:=33};
  }

  covergroup FSM_G @(posedge clk);

    label1 : coverpoint x {
      bins t_x = (0 => 1 => 0);
    }

  endgroup

  function new();
      FSM_G = new;

  endfunction
endclass

endpackage

```



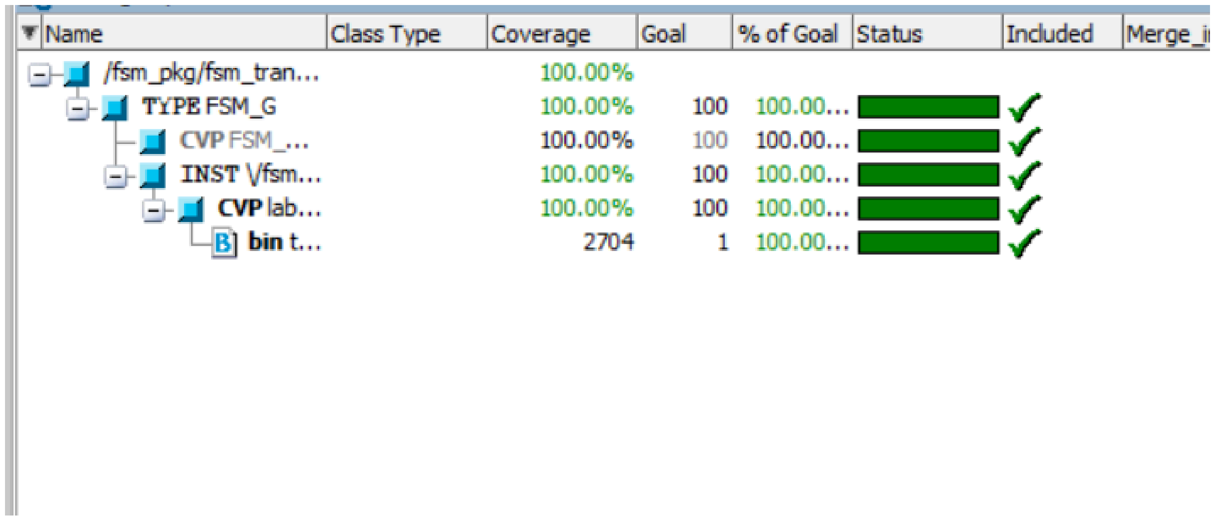
```
initial begin
  clk = 0;
  forever begin
    #1 clk = ~clk;
    my_fsm1.clk = clk;
  end
end
```

DISPLAY :



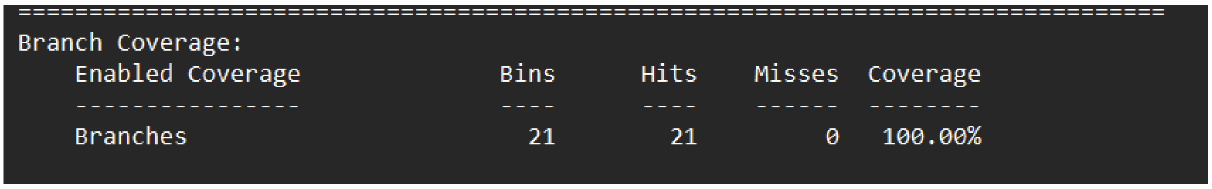
```
# Loading work.fsm_tb(fast)
# Loading work.FSM_010(fast)
# error =            0 correct =       18001
# ** Note: $stop    : fsm_tb.sv(45)
#    Time: 36002 ns  Iteration: 1  Instance: /fsm_tb
# Break in Module fsm_tb at fsm_tb.sv line 45
```

COVERGROUP :

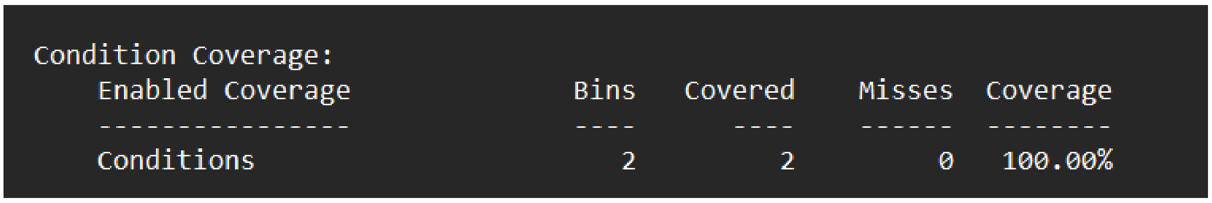


Name	Class Type	Coverage	Goal	% of Goal	Status	Included	Merge_i
/fsm_pkg/fsm_tran...		100.00%					
TYPE FSM_G		100.00%	100	100.00...	🟩	✓	
CVP FSM_...		100.00%	100	100.00...	🟩	✓	
INST \/fsm...		100.00%	100	100.00...	🟩	✓	
CVP lab...		100.00%	100	100.00...	🟩	✓	
bin t...		2704	1	100.00...	🟩	✓	

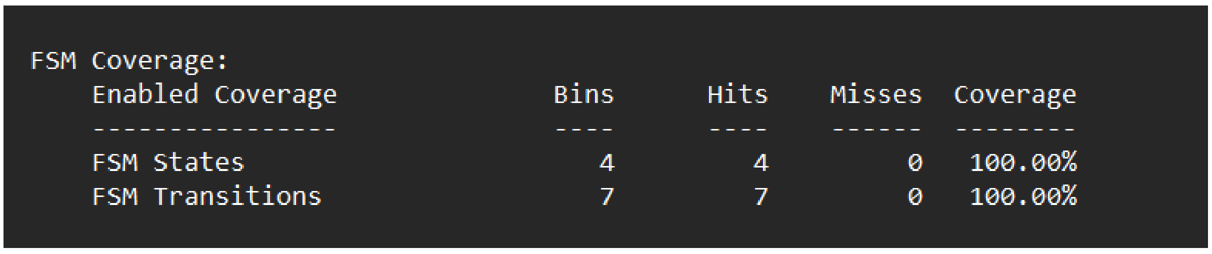
CODE COVERAGE :



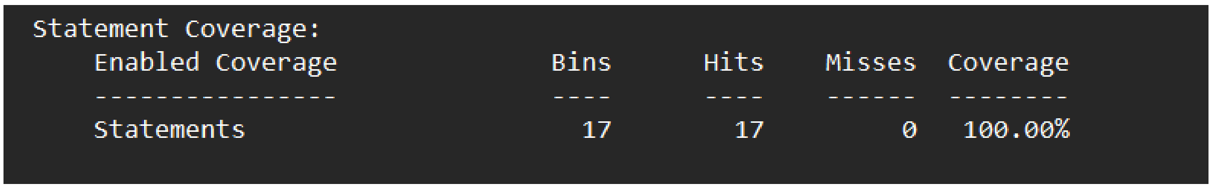
```
===============================================================================
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        21        21         0   100.00%
```



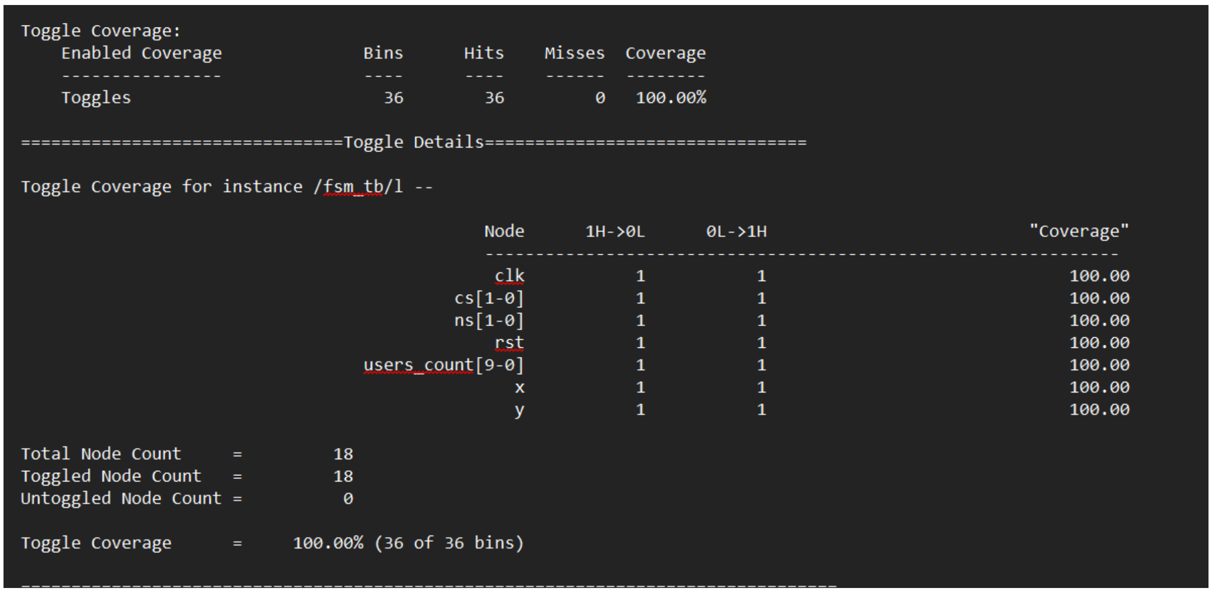
```
Condition Coverage:
    Enabled Coverage              Bins   Covered    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Conditions                       2         2         0   100.00%
```



```
FSM Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    FSM States                       4         4         0   100.00%
    FSM Transitions                  7         7         0   100.00%
```



```
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                      17        17         0   100.00%
```



```
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         36        36         0   100.00%

================================Toggle Details================================

Toggle Coverage for instance /fsm_tb/l --

                                              Node      1H->0L      0L->1H                          "Coverage"
                                              ---------------------------------------------------------------
                                               clk           1           1                              100.00
                                           cs[1-0]           1           1                              100.00
                                           ns[1-0]           1           1                              100.00
                                               rst           1           1                              100.00
                                  users_count[9-0]           1           1                              100.00
                                                 x           1           1                              100.00
                                                 y           1           1                              100.00

Total Node Count     =         18
Toggled Node Count   =         18
Untoggled Node Count =          0

Toggle Coverage      =     100.00% (36 of 36 bins)

-----------------------------------------------------------------------------------
```