MARIAM MOHAMED MARZOUK

ASSIGNMENT_3

Q2. Verify the functionality of the following counter:

- **Parameters:**
  1. WIDTH: width of the data_load and count_out ports (Valid values: 4, 6, 8, default: 4)
- **Inputs**:
  1. clk
  2. rst_n (active low sync rst)
  3. load_n (active low load)
  4. up_down (When the input is high then increment counter, else decrement the counter)
  5. ce (enable signal to increment or decrement the counter depending on the up_down)
  6. data_load (load data to count_out output when the load signal is asserted)
- **Outputs**:
  1. count_out (counter output)
  2. max_count (When the counter reaches the maximum value, this signal is high, else low)
  3. zero (When the counter reaches the minimum value, this signal is high, else low)

Requirements:

1. Create a verification plan document based on your verification plan items to support your verification planning, an example of the document can be found in the link here. Please copy this document to have your own version
2. Create a package that has a class with the following constraints

   a. Constraint the reset to be deactivated most of the time
   b. Constraint the load signal to be active 70% of the time
   c. Constraint the enable signal to be active 70% of the time
3. Create a testbench that randomize the data in a repeat block or for loop using the class object to use the above constraints. Make the testbench self-checking.

You are free to add more constraints to enrich your verification to reach 100% code coverage.

Question 1

I. Verification plan

Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
COUNTER_1	When reset (rst_n) is asserted, the counter output count_out should reset to zero.	Randomize rst_n signal, and apply during the simulation to test the reset functionality.	Ensure transitions of rst_n are covered, and the reset condition is observed.	Check if count_out is set to zero when rst_n is low.
COUNTER_2	When load_n is asserted, the counter should load the value of data_load into count_out.	Randomized values for load_n and data_load are generated and applied.	Ensure all values of data_load are covered, and the load operation is validated.	A checker verifies if count_out matches data_load when load_n is low.
COUNTER_3	When up_down is high, the counter should increment; when up_down is low, the counter should decrement.	Randomize up_down signal and apply during the test to simulate both increment and decrement operations.	Ensure functional coverage for both increment and decrement scenarios, including boundary conditions.	A checker verifies the correct increment and decrement of count_out based on the value of up_down.
COUNTER_4	The counter should wrap around to zero after reaching its maximum value, and wrap to max value after reaching zero (underflow).	Randomize ce, up_down, and allow count_out to increment or decrement beyond the boundary conditions.	Validate functional coverage for both overflow and underflow conditions.	A checker ensures count_out wraps around to zero when it reaches the maximum value, and to max value when decrementing from zero.
COUNTER_5	When ce (count enable) is low, the counter should not increment or decrement, and count_out should remain constant.	Randomize ce signal to enable and disable counting operations during the test.	Ensure all scenarios for counting enabled and disabled conditions are covered.	A checker ensures count_out remains constant when ce is low and increments or decrements when ce is high.
COUNTER_6	Correctness of max_count and zero flags based on the counter's state.	Randomize count_out to test the flags when the counter is at maximum value or zero.	Ensure functional coverage for max_count and zero flag transitions.	A checker ensures the max_count flag is set when the counter reaches its maximum value, and the zero flag when it reaches zero.

II. RTL Design

```
////////////////////////////////////////////////////////////////////////////////
// Author: Kareem Waseem
// Course: Digital Verification using SV & UVM
//
// Description: Counter Design
//
////////////////////////////////////////////////////////////////////////////////
module counter (clk ,rst_n, load_n, up_down, ce, data_load, count_out, max_count, zero);
parameter WIDTH = 4;
input clk;
input rst_n;
input load_n;
input up_down;
input ce;
input [WIDTH-1:0] data_load;
output reg [WIDTH-1:0] count_out;
output max_count;
output zero;

always @(posedge clk) begin
    if (!rst_n)
        count_out <= 0;
    else if (!load_n)
        count_out <= data_load;
    else if (ce)
        if (up_down)
            count_out <= count_out + 1;
        else
            count_out <= count_out - 1;
end

assign max_count = (count_out == {WIDTH{1'b1}})? 1:0;
assign zero = (count_out == 0)? 1:0;

endmodule
```

III. Print

```
DUT Output: count_out = 3, Expected Output: count_out_exp = 3
DUT Output: count_out = 13, Expected Output: count_out_exp = 13
DUT Output: count_out = 10, Expected Output: count_out_exp = 10
DUT Output: count_out = 5, Expected Output: count_out_exp = 5
DUT Output: count_out = 12, Expected Output: count_out_exp = 12
DUT Output: count_out = 13, Expected Output: count_out_exp = 13
DUT Output: count_out = 4, Expected Output: count_out_exp = 4
DUT Output: count_out = 3, Expected Output: count_out_exp = 3
DUT Output: count_out = 4, Expected Output: count_out_exp = 4
DUT Output: count_out = 12, Expected Output: count_out_exp = 12
DUT Output: count_out = 3, Expected Output: count_out_exp = 3
DUT Output: count_out = 3, Expected Output: count_out_exp = 3
DUT Output: count_out = 3, Expected Output: count_out_exp = 3
DUT Output: count_out = 2, Expected Output: count_out_exp = 2
DUT Output: count_out = 1, Expected Output: count_out_exp = 1
DUT Output: count_out = 4, Expected Output: count_out_exp = 4
DUT Output: count_out = 4, Expected Output: count_out_exp = 4
DUT Output: count_out = 1, Expected Output: count_out_exp = 1
DUT Output: count_out = 13, Expected Output: count_out_exp = 13
DUT Output: count_out = 14, Expected Output: count_out_exp = 14
DUT Output: count_out = 0, Expected Output: count_out_exp = 0
DUT Output: count_out = 0, Expected Output: count_out_exp = 0
DUT Output: count_out = 0, Expected Output: count_out_exp = 0
DUT Output: count_out = 1, Expected Output: count_out_exp = 1
DUT Output: count_out = 5, Expected Output: count_out_exp = 5
DUT Output: count_out = 4, Expected Output: count_out_exp = 4
DUT Output: count_out = 4, Expected Output: count_out_exp = 4
DUT Output: count_out = 8, Expected Output: count_out_exp = 8
DUT Output: count_out = 6, Expected Output: count_out_exp = 6
DUT Output: count_out = 10, Expected Output: count_out_exp = 10
DUT Output: count_out = 9, Expected Output: count_out_exp = 9
DUT Output: count_out = 5, Expected Output: count_out_exp = 5
DUT Output: count_out = 0, Expected Output: count_out_exp = 0
DUT Output: count_out = 0, Expected Output: count_out_exp = 0
DUT Output: count_out = 15, Expected Output: count_out_exp = 15
DUT Output: count_out = 15, Expected Output: count_out_exp = 15
DUT Output: count_out = 7, Expected Output: count_out_exp = 7
DUT Output: count_out = 12, Expected Output: count_out_exp = 12
DUT Output: count_out = 12, Expected Output: count_out_exp = 12
DUT Output: count_out = 12, Expected Output: count_out_exp = 12
DUT Output: count_out = 13, Expected Output: count_out_exp = 13
DUT Output: count_out = 0, Expected Output: count_out_exp = 0
DUT Output: count_out = 5, Expected Output: count_out_exp = 5
DUT Output: count_out = 6, Expected Output: count_out_exp = 6
DUT Output: count_out = 9, Expected Output: count_out_exp = 9
DUT Output: count_out = 8, Expected Output: count_out_exp = 8
```

```
# DUT Output: count_out = 6, Expected Output: count_out_exp = 6
# DUT Output: count_out = 10, Expected Output: count_out_exp = 10
# DUT Output: count_out = 4, Expected Output: count_out_exp = 4
# DUT Output: count_out = 5, Expected Output: count_out_exp = 5
# DUT Output: count_out = 10, Expected Output: count_out_exp = 10
# DUT Output: count_out = 2, Expected Output: count_out_exp = 2
# DUT Output: count_out = 3, Expected Output: count_out_exp = 3
# DUT Output: count_out = 12, Expected Output: count_out_exp = 12
# DUT Output: count_out = 1, Expected Output: count_out_exp = 1
# DUT Output: count_out = 13, Expected Output: count_out_exp = 13
# DUT Output: count_out = 14, Expected Output: count_out_exp = 14
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 1, Expected Output: count_out_exp = 1
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 11, Expected Output: count_out_exp = 11
# DUT Output: count_out = 4, Expected Output: count_out_exp = 4
# DUT Output: count_out = 8, Expected Output: count_out_exp = 8
# DUT Output: count_out = 8, Expected Output: count_out_exp = 8
# DUT Output: count_out = 3, Expected Output: count_out_exp = 3
# DUT Output: count_out = 3, Expected Output: count_out_exp = 3
# DUT Output: count_out = 4, Expected Output: count_out_exp = 4
# DUT Output: count_out = 10, Expected Output: count_out_exp = 10
# DUT Output: count_out = 10, Expected Output: count_out_exp = 10
# DUT Output: count_out = 10, Expected Output: count_out_exp = 10
# DUT Output: count_out = 10, Expected Output: count_out_exp = 10
# DUT Output: count_out = 12, Expected Output: count_out_exp = 12
# DUT Output: count_out = 12, Expected Output: count_out_exp = 12
# DUT Output: count_out = 13, Expected Output: count_out_exp = 13
# DUT Output: count_out = 14, Expected Output: count_out_exp = 14
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 1, Expected Output: count_out_exp = 1
# DUT Output: count_out = 4, Expected Output: count_out_exp = 4
# DUT Output: count_out = 6, Expected Output: count_out_exp = 6
# DUT Output: count_out = 6, Expected Output: count_out_exp = 6
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 15, Expected Output: count_out_exp = 15
# DUT Output: count_out = 14, Expected Output: count_out_exp = 14
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 15, Expected Output: count_out_exp = 15
# DUT Output: count_out = 2, Expected Output: count_out_exp = 2
# DUT Output: count_out = 13, Expected Output: count_out_exp = 13
# DUT Output: count_out = 1, Expected Output: count_out_exp = 1
# DUT Output: count_out = 5, Expected Output: count_out_exp = 5
```

```
# DUT Output: count_out = 6, Expected Output: count_out_exp = 6
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 11, Expected Output: count_out_exp = 11
# DUT Output: count_out = 11, Expected Output: count_out_exp = 11
# DUT Output: count_out = 10, Expected Output: count_out_exp = 10
# DUT Output: count_out = 14, Expected Output: count_out_exp = 14
# DUT Output: count_out = 5, Expected Output: count_out_exp = 5
# DUT Output: count_out = 6, Expected Output: count_out_exp = 6
# DUT Output: count_out = 6, Expected Output: count_out_exp = 6
# DUT Output: count_out = 2, Expected Output: count_out_exp = 2
# DUT Output: count_out = 3, Expected Output: count_out_exp = 3
# DUT Output: count_out = 9, Expected Output: count_out_exp = 9
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 6, Expected Output: count_out_exp = 6
# DUT Output: count_out = 6, Expected Output: count_out_exp = 6
# DUT Output: count_out = 11, Expected Output: count_out_exp = 11
# DUT Output: count_out = 11, Expected Output: count_out_exp = 11
# DUT Output: count_out = 12, Expected Output: count_out_exp = 12
# DUT Output: count_out = 7, Expected Output: count_out_exp = 7
# DUT Output: count_out = 7, Expected Output: count_out_exp = 7
# DUT Output: count_out = 12, Expected Output: count_out_exp = 12
# DUT Output: count_out = 4, Expected Output: count_out_exp = 4
# DUT Output: count_out = 4, Expected Output: count_out_exp = 4
# DUT Output: count_out = 0, Expected Output: count_out_exp = 0
# DUT Output: count_out = 15, Expected Output: count_out_exp = 15
# DUT Output: count_out = 13, Expected Output: count_out_exp = 13
# DUT Output: count_out = 3, Expected Output: count_out_exp = 3
# DUT Output: count_out = 1, Expected Output: count_out_exp = 1
# DUT Output: count_out = 5, Expected Output: count_out_exp = 5
# DUT Output: count_out = 5, Expected Output: count_out_exp = 5
# DUT Output: count_out = 13, Expected Output: count_out_exp = 13
# DUT Output: count_out = 2, Expected Output: count_out_exp = 2
# DUT Output: count_out = 3, Expected Output: count_out_exp = 3
# DUT Output: count_out = 2, Expected Output: count_out_exp = 2
# DUT Output: count_out = 1, Expected Output: count_out_exp = 1
# DUT Output: count_out = 12, Expected Output: count_out_exp = 12
# Total Correct Count = 1000, Total Errors = 0
```

IV. Package

```
package PCK;
  parameter WIDTH = 4;
  localparam ZERO = 0;
  localparam MAX_COUNT = {WIDTH{1'b1}};

  class E2;
    rand bit rst_n_c, load_n_c, ce_c, up_down_c;
    rand bit [WIDTH-1:0] data_load_c;
    logic [WIDTH-1:0] count_out_c;

    // Constraints for the random variables
    constraint c_rst {
      rst_n_c dist {
        0 := 5,
        1 := 95
      };
    }
    constraint c_load {
      load_n_c dist {
        0 := 70,
        1 := 30
      };
    }
    constraint c_enable {
      ce_c dist {
        0 := 30,
        1 := 70
      };
    }
    covergroup cg;
      load_cp: coverpoint data_load_c iff (!load_n_c);
      count_up_c1: coverpoint count_out_c iff (rst_n_c && ce_c && up_down_c);
      count_up_c2: coverpoint count_out_c iff (rst_n_c && ce_c && up_down_c) {
        bins trans_overflow = (MAX_COUNT => ZERO);
      }
      count_down_c1: coverpoint count_out_c iff (rst_n_c && ce_c && !up_down_c);
      count_down_c2: coverpoint count_out_c iff (rst_n_c && ce_c && !up_down_c) {
        bins trans_underflow = (ZERO => MAX_COUNT);
      }
    endgroup
    function new;
      cg = new();
    endfunction


  endclass
endpackage
```

V. Test bench

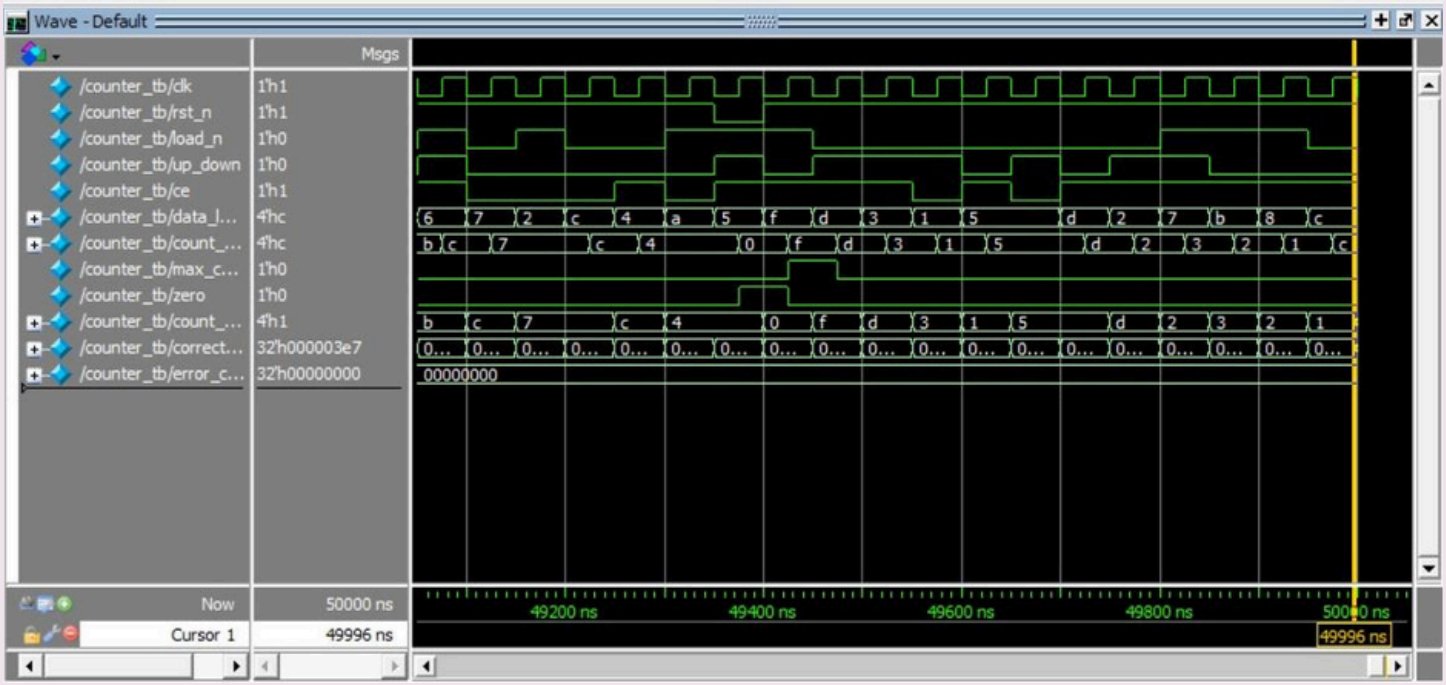
```
import PCK::*;
module counter_tb;
  // Parameter
  parameter WIDTH = 4;
  // Input and output declaration
  logic clk, rst_n, load_n, up_down, ce;
  logic [WIDTH-1:0] data_load;
  logic [WIDTH-1:0] count_out;
  logic max_count, zero;
  logic [WIDTH-1:0] count_out_exp;  // Expected value for output count
  // Counters declaration
  int correct_count, error_count;
  // DUT Instantiation
  counter #(
      .WIDTH(WIDTH)
  ) DUT (
      .clk(clk),
      .rst_n(rst_n),
      .load_n(load_n),
      .up_down(up_down),
      .ce(ce),
      .data_load(data_load),
      .count_out(count_out),
      .max_count(max_count),
      .zero(zero)
  );
  // Clock generation
  initial begin
    clk = 0;
    forever #25 clk = ~clk;  // Clock period is 50 time units
  end
  // Define the randomization object
  E2 object1;
  initial begin
    // Initialize variables
    object1 = new();
    correct_count = 0;
    error_count = 0;
    count_out_exp = 0;
    // Run the test for a number of iterations
    repeat (1000) begin
      // Randomize the test object
      assert (object1.randomize());
      // Apply randomized values
      rst_n = object1.rst_n_c;
      load_n = object1.load_n_c;
      up_down = object1.up_down_c;
      ce = object1.ce_c;
      data_load = object1.data_load_c;
      @(negedge clk);
      object1.count_out_c = count_out;
      object1.cg.sample();
      self_check();
    end
```

```
    // Final results
    $display("Total Correct Count = %0d, Total Errors = %0d", correct_count, error_count);
    $stop;
  end
  // Task to check the DUT output
  task self_check();
    // Calculate the expected value
    if (!rst_n) begin
      count_out_exp = 0;
    end else if (!load_n) begin
      count_out_exp = data_load;
    end else if (ce) begin
      // Update count_out_exp based on direction
      if (up_down) begin
        if (count_out_exp == {WIDTH{1'b1}}) count_out_exp = 0;  // Wrap around
        else count_out_exp = count_out_exp + 1;
      end else begin
        if (count_out_exp == 0) count_out_exp = {WIDTH{1'b1}};  // Wrap around
        else count_out_exp = count_out_exp - 1;
      end
    end
    // Display
    $display("DUT Output: count_out = %0d, Expected Output: count_out_exp = %0d", count_out,
             count_out_exp);
    if (count_out !== count_out_exp) begin
      $display("Error: Expected count_out = %0d, but got count_out = %0d", count_out_exp,
               count_out);
      error_count++;
    end else begin
      correct_count++;
    end
  endtask
endmodule
```

VI. Do file

```
vlib work
vlog counter.v package.svh counter_tb.svh +cover -covercells
vsim -voptargs=+acc work.counter_tb -cover
add wave *
coverage save counter_tb.ucdb -onexit
run -all
```

VII. Waveform



VIII. Functional Coverage Report

```
Coverage Report by instance with details

================================================================================
=== Instance: /PCK
=== Design Unit: work.PCK
================================================================================

Covergroup Coverage:
    Covergroups                 1        na        na   100.00%
        Coverpoints/Crosses     5        na        na        na
            Covergroup Bins    50        50         0   100.00%
--------------------------------------------------------------------------------------------------------
Covergroup                                          Metric       Goal       Bins    Status

--------------------------------------------------------------------------------------------------------
 TYPE /PCK/E2/cg                                  100.00%        100          -    Covered
    covered/total bins:                                50         50          -
    missing/total bins:                                 0         50          -
    % Hit:                                         100.00%        100          -
    Coverpoint load_cp                             100.00%        100          -    Covered
        covered/total bins:                            16         16          -
        missing/total bins:                             0         16          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_up_c1                         100.00%        100          -    Covered
        covered/total bins:                            16         16          -
        missing/total bins:                             0         16          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_up_c2                         100.00%        100          -    Covered
        covered/total bins:                             1          1          -
        missing/total bins:                             0          1          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_down_c1                       100.00%        100          -    Covered
        covered/total bins:                            16         16          -
        missing/total bins:                             0         16          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_down_c2                       100.00%        100          -    Covered
        covered/total bins:                             1          1          -
        missing/total bins:                             0          1          -
        % Hit:                                     100.00%        100          -
 Covergroup instance \/PCK::E2::cg                 100.00%        100          -    Covered
    covered/total bins:                                50         50          -
    missing/total bins:                                 0         50          -
    % Hit:                                         100.00%        100          -
    Coverpoint load_cp                             100.00%        100          -    Covered
        covered/total bins:                            16         16          -
        missing/total bins:                             0         16          -
        % Hit:                                     100.00%        100          -
        bin auto[0]                                    50          1          -    Covered
        bin auto[1]                                    36          1          -    Covered
        bin auto[2]                                    43          1          -    Covered
        bin auto[3]                                    48          1          -    Covered
        bin auto[4]                                    50          1          -    Covered
        bin auto[5]                                    48          1          -    Covered
        bin auto[6]                                    50          1          -    Covered
        bin auto[7]                                    42          1          -    Covered
        bin auto[8]                                    40          1          -    Covered
        bin auto[9]                                    50          1          -    Covered
        bin auto[10]                                   47          1          -    Covered
        bin auto[11]                                   45          1          -    Covered
        bin auto[12]                                   44          1          -    Covered
        bin auto[13]                                   42          1          -    Covered
        bin auto[14]                                   35          1          -    Covered
```

```
        bin auto[15]                                45          1          -    Covered
    Coverpoint count_up_c1                      100.00%       100          -    Covered
        covered/total bins:                          16        16          -
        missing/total bins:                           0        16          -
        % Hit:                                  100.00%       100          -
        bin auto[0]                                  18          1          -    Covered
        bin auto[1]                                  25          1          -    Covered
        bin auto[2]                                  18          1          -    Covered
        bin auto[3]                                  26          1          -    Covered
        bin auto[4]                                  26          1          -    Covered
        bin auto[5]                                  18          1          -    Covered
        bin auto[6]                                  26          1          -    Covered
        bin auto[7]                                  23          1          -    Covered
        bin auto[8]                                  21          1          -    Covered
        bin auto[9]                                  27          1          -    Covered
        bin auto[10]                                 24          1          -    Covered
        bin auto[11]                                 17          1          -    Covered
        bin auto[12]                                 23          1          -    Covered
        bin auto[13]                                 28          1          -    Covered
        bin auto[14]                                 15          1          -    Covered
        bin auto[15]                                 20          1          -    Covered
    Coverpoint count_up_c2                      100.00%       100          -    Covered
        covered/total bins:                           1          1          -
        missing/total bins:                           0          1          -
        % Hit:                                  100.00%       100          -
        bin trans_overflow                            1          1          -    Covered
    Coverpoint count_down_c1                    100.00%       100          -    Covered
        covered/total bins:                          16        16          -
        missing/total bins:                           0        16          -
        % Hit:                                  100.00%       100          -
        bin auto[0]                                  23          1          -    Covered
        bin auto[1]                                  16          1          -    Covered
        bin auto[2]                                  26          1          -    Covered
        bin auto[3]                                  24          1          -    Covered
        bin auto[4]                                  15          1          -    Covered
        bin auto[5]                                  24          1          -    Covered
        bin auto[6]                                  19          1          -    Covered
        bin auto[7]                                  13          1          -    Covered
        bin auto[8]                                  18          1          -    Covered
        bin auto[9]                                  20          1          -    Covered
        bin auto[10]                                 18          1          -    Covered
        bin auto[11]                                 18          1          -    Covered
        bin auto[12]                                 22          1          -    Covered
        bin auto[13]                                 15          1          -    Covered
        bin auto[14]                                 18          1          -    Covered
        bin auto[15]                                 34          1          -    Covered
    Coverpoint count_down_c2                    100.00%       100          -    Covered
        covered/total bins:                           1          1          -
        missing/total bins:                           0          1          -
        % Hit:                                  100.00%       100          -
        bin trans_underflow                           5          1          -    Covered

COVERGROUP COVERAGE:
----------------------------------------------------------------------------------------------------
Covergroup                                      Metric       Goal       Bins    Status

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

```
----------------------------------------------------------------------------------------------------
TYPE /PCK/E2/cg                                  100.00%        100          -    Covered
   covered/total bins:                                50         50          -
   missing/total bins:                                 0         50          -
   % Hit:                                         100.00%        100          -
   Coverpoint load_cp                             100.00%        100          -    Covered
       covered/total bins:                            16         16          -
       missing/total bins:                             0         16          -
       % Hit:                                     100.00%        100          -
   Coverpoint count_up_c1                         100.00%        100          -    Covered
       covered/total bins:                            16         16          -
       missing/total bins:                             0         16          -
       % Hit:                                     100.00%        100          -
   Coverpoint count_up_c2                         100.00%        100          -    Covered
       covered/total bins:                             1          1          -
       missing/total bins:                             0          1          -
       % Hit:                                     100.00%        100          -
   Coverpoint count_down_c1                       100.00%        100          -    Covered
       covered/total bins:                            16         16          -
       missing/total bins:                             0         16          -
       % Hit:                                     100.00%        100          -
   Coverpoint count_down_c2                       100.00%        100          -    Covered
       covered/total bins:                             1          1          -
       missing/total bins:                             0          1          -
       % Hit:                                     100.00%        100          -
Covergroup instance \/PCK::E2::cg                100.00%        100          -    Covered
   covered/total bins:                                50         50          -
   missing/total bins:                                 0         50          -
   % Hit:                                         100.00%        100          -
   Coverpoint load_cp                             100.00%        100          -    Covered
       covered/total bins:                            16         16          -
       missing/total bins:                             0         16          -
       % Hit:                                     100.00%        100          -
       bin auto[0]                                    50          1          -    Covered
       bin auto[1]                                    36          1          -    Covered
       bin auto[2]                                    43          1          -    Covered
       bin auto[3]                                    48          1          -    Covered
       bin auto[4]                                    50          1          -    Covered
       bin auto[5]                                    48          1          -    Covered
       bin auto[6]                                    50          1          -    Covered
       bin auto[7]                                    42          1          -    Covered
       bin auto[8]                                    40          1          -    Covered
       bin auto[9]                                    50          1          -    Covered
       bin auto[10]                                   47          1          -    Covered
       bin auto[11]                                   45          1          -    Covered
       bin auto[12]                                   44          1          -    Covered
       bin auto[13]                                   42          1          -    Covered
       bin auto[14]                                   35          1          -    Covered
       bin auto[15]                                   45          1          -    Covered
   Coverpoint count_up_c1                         100.00%        100          -    Covered
       covered/total bins:                            16         16          -
       missing/total bins:                             0         16          -
       % Hit:                                     100.00%        100          -
       bin auto[0]                                    18          1          -    Covered
       bin auto[1]                                    25          1          -    Covered
       bin auto[2]                                    18          1          -    Covered
       bin auto[3]                                    26          1          -    Covered
       bin auto[4]                                    26          1          -    Covered
       bin auto[5]                                    18          1          -    Covered
       bin auto[6]                                    26          1          -    Covered
       bin auto[7]                                    23          1          -    Covered
       bin auto[8]                                    21          1          -    Covered
       bin auto[9]                                    27          1          -    Covered
```

```
        bin auto[8]                                40          1          -    Covered
        bin auto[9]                                50          1          -    Covered
        bin auto[10]                               47          1          -    Covered
        bin auto[11]                               45          1          -    Covered
        bin auto[12]                               44          1          -    Covered
        bin auto[13]                               42          1          -    Covered
        bin auto[14]                               35          1          -    Covered
        bin auto[15]                               45          1          -    Covered
    Coverpoint count_up_c1                    100.00%        100          -    Covered
        covered/total bins:                        16         16          -
        missing/total bins:                         0         16          -
        % Hit:                                100.00%        100          -
        bin auto[0]                                18          1          -    Covered
        bin auto[1]                                25          1          -    Covered
        bin auto[2]                                18          1          -    Covered
        bin auto[3]                                26          1          -    Covered
        bin auto[4]                                26          1          -    Covered
        bin auto[5]                                18          1          -    Covered
        bin auto[6]                                26          1          -    Covered
        bin auto[7]                                23          1          -    Covered
        bin auto[8]                                21          1          -    Covered
        bin auto[9]                                27          1          -    Covered
        bin auto[10]                               24          1          -    Covered
        bin auto[11]                               17          1          -    Covered
        bin auto[12]                               23          1          -    Covered
        bin auto[13]                               28          1          -    Covered
        bin auto[14]                               15          1          -    Covered
        bin auto[15]                               20          1          -    Covered
    Coverpoint count_up_c2                    100.00%        100          -    Covered
        covered/total bins:                         1          1          -
        missing/total bins:                         0          1          -
        % Hit:                                100.00%        100          -
        bin trans_overflow                          1          1          -    Covered
    Coverpoint count_down_c1                  100.00%        100          -    Covered
        covered/total bins:                        16         16          -
        missing/total bins:                         0         16          -
        % Hit:                                100.00%        100          -
        bin auto[0]                                23          1          -    Covered
        bin auto[1]                                16          1          -    Covered
        bin auto[2]                                26          1          -    Covered
        bin auto[3]                                24          1          -    Covered
        bin auto[4]                                15          1          -    Covered
        bin auto[5]                                24          1          -    Covered
        bin auto[6]                                19          1          -    Covered
        bin auto[7]                                13          1          -    Covered
        bin auto[8]                                18          1          -    Covered
        bin auto[9]                                20          1          -    Covered
        bin auto[10]                               18          1          -    Covered
        bin auto[11]                               18          1          -    Covered
        bin auto[12]                               22          1          -    Covered
        bin auto[13]                               15          1          -    Covered
        bin auto[14]                               18          1          -    Covered
        bin auto[15]                               34          1          -    Covered
    Coverpoint count_down_c2                  100.00%        100          -    Covered
        covered/total bins:                         1          1          -
        missing/total bins:                         0          1          -
        % Hit:                                100.00%        100          -
        bin trans_underflow                         5          1          -    Covered

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1

Total Coverage By Instance (filtered view): 100.00%
```

```
        bin auto[8]                              40          1          -    Covered
        bin auto[9]                              50          1          -    Covered
        bin auto[10]                             47          1          -    Covered
        bin auto[11]                             45          1          -    Covered
        bin auto[12]                             44          1          -    Covered
        bin auto[13]                             42          1          -    Covered
        bin auto[14]                             35          1          -    Covered
        bin auto[15]                             45          1          -    Covered
    Coverpoint count_up_c1                  100.00%        100          -    Covered
        covered/total bins:                      16         16          -
        missing/total bins:                       0         16          -
        % Hit:                              100.00%        100          -
        bin auto[0]                              18          1          -    Covered
        bin auto[1]                              25          1          -    Covered
        bin auto[2]                              18          1          -    Covered
        bin auto[3]                              26          1          -    Covered
        bin auto[4]                              26          1          -    Covered
        bin auto[5]                              18          1          -    Covered
        bin auto[6]                              26          1          -    Covered
        bin auto[7]                              23          1          -    Covered
        bin auto[8]                              21          1          -    Covered
        bin auto[9]                              27          1          -    Covered
        bin auto[10]                             24          1          -    Covered
        bin auto[11]                             17          1          -    Covered
        bin auto[12]                             23          1          -    Covered
        bin auto[13]                             28          1          -    Covered
        bin auto[14]                             15          1          -    Covered
        bin auto[15]                             20          1          -    Covered
    Coverpoint count_up_c2                  100.00%        100          -    Covered
        covered/total bins:                       1          1          -
        missing/total bins:                       0          1          -
        % Hit:                              100.00%        100          -
        bin trans_overflow                        1          1          -    Covered
    Coverpoint count_down_c1                100.00%        100          -    Covered
        covered/total bins:                      16         16          -
        missing/total bins:                       0         16          -
        % Hit:                              100.00%        100          -
        bin auto[0]                              23          1          -    Covered
        bin auto[1]                              16          1          -    Covered
        bin auto[2]                              26          1          -    Covered
        bin auto[3]                              24          1          -    Covered
        bin auto[4]                              15          1          -    Covered
        bin auto[5]                              24          1          -    Covered
        bin auto[6]                              19          1          -    Covered
        bin auto[7]                              13          1          -    Covered
        bin auto[8]                              18          1          -    Covered
        bin auto[9]                              20          1          -    Covered
        bin auto[10]                             18          1          -    Covered
        bin auto[11]                             18          1          -    Covered
        bin auto[12]                             22          1          -    Covered
        bin auto[13]                             15          1          -    Covered
        bin auto[14]                             18          1          -    Covered
        bin auto[15]                             34          1          -    Covered
    Coverpoint count_down_c2                100.00%        100          -    Covered
        covered/total bins:                       1          1          -
        missing/total bins:                       0          1          -
        % Hit:                              100.00%        100          -
        bin trans_underflow                       5          1          -    Covered

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1

Total Coverage By Instance (filtered view): 100.00%
```

IX. Coverage report

```
Coverage Report by instance with details

================================================================================
=== Instance: /counter_tb/DUT
=== Design Unit: work.counter
================================================================================
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        10        10         0   100.00%

================================Branch Details================================

Branch Coverage for instance /counter_tb/DUT

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File counter.v
------------------------------------IF Branch------------------------------------
    21                                       999     Count coming in to IF
    21              1                         61
    23              1                        662
    25              1                        204
                                              72     All False Count
Branch totals: 4 hits of 4 branches = 100.00%

------------------------------------IF Branch------------------------------------
    26                                       204     Count coming in to IF
    26              1                        107
    28              1                         97
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    32                                       871     Count coming in to IF
    32              1                         65
    32              2                        806
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    33                                       871     Count coming in to IF
    33              1                        102
    33              2                        769
Branch totals: 2 hits of 2 branches = 100.00%


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

================================Condition Details================================

Condition Coverage for instance /counter_tb/DUT --

  File counter.v
----------------Focused Condition View-------------------
Line       32 Item    1  (count_out == {4{1}})
Condition totals: 1 of 1 input term covered = 100.00%

               Input Term   Covered  Reason for no coverage   Hint
              -----------  --------  ----------------------  --------------
  (count_out == {4{1}})         Y

     Rows:       Hits  FEC Target                Non-masking condition(s)
 ---------  ---------  --------------------      -------------------------
  Row   1:          1  (count_out == {4{1}})_0  -
  Row   2:          1  (count_out == {4{1}})_1  -

----------------Focused Condition View-------------------
Line       33 Item    1  (count_out == 0)
Condition totals: 1 of 1 input term covered = 100.00%

        Input Term   Covered  Reason for no coverage   Hint
       -----------  --------  ----------------------  --------------
  (count_out == 0)         Y
```

```
        Input Term   Covered  Reason for no coverage   Hint
       -----------  --------  ----------------------  --------------
  (count_out == 0)         Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  (count_out == 0)_0    -
  Row   2:          1  (count_out == 0)_1    -


Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                       7         7         0   100.00%

================================Statement Details================================

Statement Coverage for instance /counter_tb/DUT --

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File counter.v
    20              1                        999
    22              1                         61
    24              1                        662
    27              1                        107
    29              1                         97
    32              1                        872
    33              1                        872
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         30        30         0   100.00%

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

Toggle Coverage for instance /counter_tb/DUT --

                                          Node      1H->0L      0L->1H  "Coverage"
                                          ----------------------------------------
                                            ce           1           1      100.00
                                           clk           1           1      100.00
                                 count_out[3-0]          1           1      100.00
                                 data_load[0-3]          1           1      100.00
                                        load_n           1           1      100.00
                                     max_count           1           1      100.00
                                         rst_n           1           1      100.00
                                       up_down           1           1      100.00
                                          zero           1           1      100.00

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

Toggle Coverage      =     100.00% (30 of 30 bins)

================================================================================
=== Instance: /counter_tb
=== Design Unit: work.counter_tb
================================================================================

Assertion Coverage:
    Assertions                       1         1         0   100.00%
--------------------------------------------------------------------
Name                 File(Line)                   Failure      Pass
                                                  Count        Count
--------------------------------------------------------------------
/counter_tb/#ublk#95084642#39/immed__41
                     counter_tb.svh(41)                 0          1
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        12        11         1    91.66%
```

```
================================Branch Details================================

Branch Coverage for instance /counter_tb

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File counter_tb.svh
------------------------------------IF Branch------------------------------------
    60                                      1000     Count coming in to IF
    60              1                         61
    62              1                        663
    64              1                        204
                                              72     All False Count
Branch totals: 4 hits of 4 branches = 100.00%

------------------------------------IF Branch------------------------------------
    66                                       204     Count coming in to IF
    66              1                        107
    71              1                         97
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    67                                       107     Count coming in to IF
    67              1                          4
    69              1                        103
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    72                                        97     Count coming in to IF
    72              1                         20
    74              1                         77
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    80                                      1000     Count coming in to IF
    80              1                     ***0***
    83              1                        1000
Branch totals: 1 hit of 2 branches = 50.00%


Condition Coverage:
    Enabled Coverage              Bins   Covered    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Conditions                       3         2         1    66.66%

================================Condition Details================================

Condition Coverage for instance /counter_tb --

  File counter_tb.svh
----------------Focused Condition View-------------------
Line       67 Item    1  (count_out_exp == {4{1}})
Condition totals: 1 of 1 input term covered = 100.00%

                Input Term   Covered  Reason for no coverage   Hint
               -----------  --------  ----------------------  --------------
  (count_out_exp == {4{1}})         Y

     Rows:       Hits  FEC Target                  Non-masking condition(s)
 ---------  ---------  --------------------        -------------------------
  Row   1:          1  (count_out_exp == {4{1}})_0  -
  Row   2:          1  (count_out_exp == {4{1}})_1  -

----------------Focused Condition View-------------------
Line       72 Item    1  (count_out_exp == 0)
Condition totals: 1 of 1 input term covered = 100.00%

            Input Term   Covered  Reason for no coverage   Hint
           -----------  --------  ----------------------  --------------
  (count_out_exp == 0)         Y

     Rows:       Hits  FEC Target              Non-masking condition(s)
 ---------  ---------  --------------------    -------------------------
  Row   1:          1  (count_out_exp == 0)_0  -
  Row   2:          1  (count_out_exp == 0)_1  -

----------------Focused Condition View-------------------
Line       80 Item    1  (count_out !== count_out_exp)
Condition totals: 0 of 1 input term covered = 0.00%
```

```
                    Input Term   Covered  Reason for no coverage   Hint
                   -----------  --------  ------------------------  --------------
  (count_out !== count_out_exp)         N  '_1' not hit             Hit '_1'

     Rows:       Hits  FEC Target                       Non-masking condition(s)
 ---------  ---------  ---------------------            -------------------------
  Row   1:          1  (count_out !== count_out_exp)_0  -
  Row   2:     ***0***  (count_out !== count_out_exp)_1  -


Statement Coverage:
    Enabled Coverage             Bins      Hits    Misses  Coverage
    ----------------             ----      ----    ------  --------
    Statements                     30        28         2    93.33%

================================Statement Details================================

Statement Coverage for instance /counter_tb --

    Line         Item                     Count     Source
    ----         ----                     -----     ------
  File counter_tb.svh
    27              1                         1
    28              1                         1
    28              2                      2001
    28              3                      2000
    34              1                         1
    35              1                         1
    36              1                         1
    37              1                         1
    39              1                      1000
    43              1                      1000
    44              1                      1000
    45              1                      1000
    46              1                      1000
    47              1                      1000
    48              1                      1000
    49              1                      1000
    50              1                      1000
    51              1                      1000
    54              1                         1
    55              1                         1
    61              1                        61
    63              1                       663
    68              1                         4
    70              1                       103
    73              1                        20
    75              1                        77
    79              1                      1000
    81              1                    ***0***
    82              1                    ***0***
    84              1                      1000
Toggle Coverage:
    Enabled Coverage             Bins      Hits    Misses  Coverage
    ----------------             ----      ----    ------  --------
    Toggles                       166        57       109    34.33%

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

Toggle Coverage for instance /counter_tb --

                                          Node      1H->0L      0L->1H  "Coverage"
                                         ----------------------------------------
                                           ce           1           1      100.00
                                          clk           1           1      100.00
                                correct_count[0-8]      1           1      100.00
                                  correct_count[9]      0           1       50.00
                               correct_count[10-31]     0           0        0.00
                                     count_out[0-3]     1           1      100.00
                                 count_out_exp[0-3]     1           1      100.00
                                    data_load[0-3]      1           1      100.00
                                 error_count[0-31]      0           0        0.00
                                           load_n       1           1      100.00
                                        max_count       1           1      100.00
                                            rst_n       1           1      100.00
                                          up_down       1           1      100.00
                                             zero       1           1      100.00
```

```
Total Node Count     =         83
Toggled Node Count   =         28
Untoggled Node Count =         55

Toggle Coverage      =      34.33% (57 of 166 bins)

================================================================================
=== Instance: /PCK
=== Design Unit: work.PCK
================================================================================

Covergroup Coverage:
    Covergroups                  1        na        na   100.00%
        Coverpoints/Crosses      5        na        na        na
            Covergroup Bins     50        50         0   100.00%
----------------------------------------------------------------------------------------------------------------
Covergroup                                          Metric       Goal       Bins    Status

----------------------------------------------------------------------------------------------------------------
 TYPE /PCK/E2/cg                                   100.00%        100          -    Covered
    covered/total bins:                                 50         50          -
    missing/total bins:                                  0         50          -
    % Hit:                                          100.00%        100          -
    Coverpoint load_cp                              100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                      100.00%        100          -
    Coverpoint count_up_c1                          100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                      100.00%        100          -
    Coverpoint count_up_c2                          100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                      100.00%        100          -
    Coverpoint count_down_c1                        100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                      100.00%        100          -
    Coverpoint count_down_c2                        100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                      100.00%        100          -
 Covergroup instance \/PCK::E2::cg                  100.00%        100          -    Covered
    covered/total bins:                                 50         50          -
    missing/total bins:                                  0         50          -
    % Hit:                                          100.00%        100          -
    Coverpoint load_cp                              100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                      100.00%        100          -
        bin auto[0]                                     50          1          -    Covered
        bin auto[1]                                     36          1          -    Covered
        bin auto[2]                                     43          1          -    Covered
        bin auto[3]                                     48          1          -    Covered
        bin auto[4]                                     50          1          -    Covered
        bin auto[5]                                     48          1          -    Covered
        bin auto[6]                                     50          1          -    Covered
        bin auto[7]                                     42          1          -    Covered
        bin auto[8]                                     40          1          -    Covered
        bin auto[9]                                     50          1          -    Covered
        bin auto[10]                                    47          1          -    Covered
        bin auto[11]                                    45          1          -    Covered
        bin auto[12]                                    44          1          -    Covered
        bin auto[13]                                    42          1          -    Covered
        bin auto[14]                                    35          1          -    Covered
        bin auto[15]                                    45          1          -    Covered
    Coverpoint count_up_c1                          100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                      100.00%        100          -
        bin auto[0]                                     18          1          -    Covered
        bin auto[1]                                     25          1          -    Covered
        bin auto[2]                                     18          1          -    Covered
        bin auto[3]                                     26          1          -    Covered
        bin auto[4]                                     26          1          -    Covered
        bin auto[5]                                     18          1          -    Covered
        bin auto[6]                                     26          1          -    Covered
```

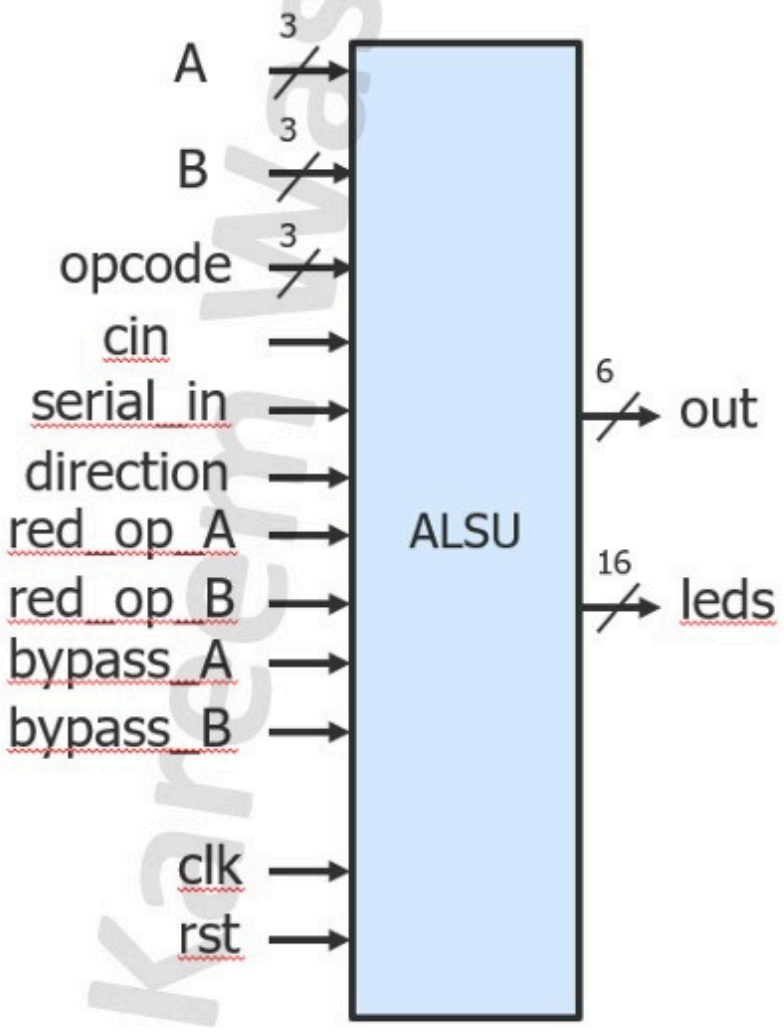
```
Total Node Count     =         83
Toggled Node Count   =         28
Untoggled Node Count =         55

Toggle Coverage      =      34.33% (57 of 166 bins)

================================================================================
=== Instance: /PCK
=== Design Unit: work.PCK
================================================================================

Covergroup Coverage:
    Covergroups                    1        na        na   100.00%
        Coverpoints/Crosses        5        na        na        na
            Covergroup Bins       50        50         0   100.00%
--------------------------------------------------------------------------------------------------------------
Covergroup                                          Metric       Goal       Bins    Status

--------------------------------------------------------------------------------------------------------------
 TYPE /PCK/E2/cg                                   100.00%        100          -    Covered
    covered/total bins:                                 50         50          -
    missing/total bins:                                  0         50          -
    % Hit:                                         100.00%        100          -
    Coverpoint load_cp                             100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_up_c1                         100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_up_c2                         100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_down_c1                       100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_down_c2                       100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                     100.00%        100          -
 Covergroup instance \/PCK::E2::cg                 100.00%        100          -    Covered
    covered/total bins:                                 50         50          -
    missing/total bins:                                  0         50          -
    % Hit:                                         100.00%        100          -
    Coverpoint load_cp                             100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                     100.00%        100          -
        bin auto[0]                                     50          1          -    Covered
        bin auto[1]                                     36          1          -    Covered
        bin auto[2]                                     43          1          -    Covered
        bin auto[3]                                     48          1          -    Covered
        bin auto[4]                                     50          1          -    Covered
        bin auto[5]                                     48          1          -    Covered
        bin auto[6]                                     50          1          -    Covered
        bin auto[7]                                     42          1          -    Covered
        bin auto[8]                                     40          1          -    Covered
        bin auto[9]                                     50          1          -    Covered
        bin auto[10]                                    47          1          -    Covered
        bin auto[11]                                    45          1          -    Covered
        bin auto[12]                                    44          1          -    Covered
        bin auto[13]                                    42          1          -    Covered
        bin auto[14]                                    35          1          -    Covered
        bin auto[15]                                    45          1          -    Covered
    Coverpoint count_up_c1                         100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                     100.00%        100          -
        bin auto[0]                                     18          1          -    Covered
        bin auto[1]                                     25          1          -    Covered
        bin auto[2]                                     18          1          -    Covered
        bin auto[3]                                     26          1          -    Covered
        bin auto[4]                                     26          1          -    Covered
        bin auto[5]                                     18          1          -    Covered
        bin auto[6]                                     26          1          -    Covered
```

```
        bin auto[7]                                23          1          -    Covered
        bin auto[8]                                21          1          -    Covered
        bin auto[9]                                27          1          -    Covered
        bin auto[10]                               24          1          -    Covered
        bin auto[11]                               17          1          -    Covered
        bin auto[12]                               23          1          -    Covered
        bin auto[13]                               28          1          -    Covered
        bin auto[14]                               15          1          -    Covered
        bin auto[15]                               20          1          -    Covered
    Coverpoint count_up_c2                     100.00%        100          -    Covered
        covered/total bins:                          1          1          -
        missing/total bins:                          0          1          -
        % Hit:                                 100.00%        100          -
        bin trans_overflow                           1          1          -    Covered
    Coverpoint count_down_c1                   100.00%        100          -    Covered
        covered/total bins:                         16         16          -
        missing/total bins:                          0         16          -
        % Hit:                                 100.00%        100          -
        bin auto[0]                                 23          1          -    Covered
        bin auto[1]                                 16          1          -    Covered
        bin auto[2]                                 26          1          -    Covered
        bin auto[3]                                 24          1          -    Covered
        bin auto[4]                                 15          1          -    Covered
        bin auto[5]                                 24          1          -    Covered
        bin auto[6]                                 19          1          -    Covered
        bin auto[7]                                 13          1          -    Covered
        bin auto[8]                                 18          1          -    Covered
        bin auto[9]                                 20          1          -    Covered
        bin auto[10]                                18          1          -    Covered
        bin auto[11]                                18          1          -    Covered
        bin auto[12]                                22          1          -    Covered
        bin auto[13]                                15          1          -    Covered
        bin auto[14]                                18          1          -    Covered
        bin auto[15]                                34          1          -    Covered
    Coverpoint count_down_c2                   100.00%        100          -    Covered
        covered/total bins:                          1          1          -
        missing/total bins:                          0          1          -
        % Hit:                                 100.00%        100          -
        bin trans_underflow                          5          1          -    Covered
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                       1         1         0   100.00%

================================Statement Details================================

Statement Coverage for instance /PCK --

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File package.svh
    25              1                          1

COVERGROUP COVERAGE:
----------------------------------------------------------------------------------------------------------------
Covergroup                                          Metric       Goal       Bins    Status

----------------------------------------------------------------------------------------------------------------
 TYPE /PCK/E2/cg                                   100.00%        100          -    Covered
    covered/total bins:                                 50         50          -
    missing/total bins:                                  0         50          -
    % Hit:                                         100.00%        100          -
    Coverpoint load_cp                             100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_up_c1                         100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_up_c2                         100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_down_c1                       100.00%        100          -    Covered
        covered/total bins:                             16         16          -
        missing/total bins:                              0         16          -
        % Hit:                                     100.00%        100          -
    Coverpoint count_down_c2                       100.00%        100          -    Covered
        covered/total bins:                              1          1          -
```

```
        % Hit:                          100.00%        100          -
        bin auto[0]                          50          1          -    Covered
        bin auto[1]                          36          1          -    Covered
        bin auto[2]                          43          1          -    Covered
        bin auto[3]                          48          1          -    Covered
        bin auto[4]                          50          1          -    Covered
        bin auto[5]                          48          1          -    Covered
        bin auto[6]                          50          1          -    Covered
        bin auto[7]                          42          1          -    Covered
        bin auto[8]                          40          1          -    Covered
        bin auto[9]                          50          1          -    Covered
        bin auto[10]                         47          1          -    Covered
        bin auto[11]                         45          1          -    Covered
        bin auto[12]                         44          1          -    Covered
        bin auto[13]                         42          1          -    Covered
        bin auto[14]                         35          1          -    Covered
        bin auto[15]                         45          1          -    Covered
    Coverpoint count_up_c1              100.00%        100          -    Covered
        covered/total bins:                  16         16          -
        missing/total bins:                   0         16          -
        % Hit:                          100.00%        100          -
        bin auto[0]                          18          1          -    Covered
        bin auto[1]                          25          1          -    Covered
        bin auto[2]                          18          1          -    Covered
        bin auto[3]                          26          1          -    Covered
        bin auto[4]                          26          1          -    Covered
        bin auto[5]                          18          1          -    Covered
        bin auto[6]                          26          1          -    Covered
        bin auto[7]                          23          1          -    Covered
        bin auto[8]                          21          1          -    Covered
        bin auto[9]                          27          1          -    Covered
        bin auto[10]                         24          1          -    Covered
        bin auto[11]                         17          1          -    Covered
        bin auto[12]                         23          1          -    Covered
        bin auto[13]                         28          1          -    Covered
        bin auto[14]                         15          1          -    Covered
        bin auto[15]                         20          1          -    Covered
    Coverpoint count_up_c2              100.00%        100          -    Covered
        covered/total bins:                   1          1          -
        missing/total bins:                   0          1          -
        % Hit:                          100.00%        100          -
        bin trans_overflow                    1          1          -    Covered
    Coverpoint count_down_c1            100.00%        100          -    Covered
        covered/total bins:                  16         16          -
        missing/total bins:                   0         16          -
        % Hit:                          100.00%        100          -
        bin auto[0]                          23          1          -    Covered
        bin auto[1]                          16          1          -    Covered
        bin auto[2]                          26          1          -    Covered
        bin auto[3]                          24          1          -    Covered
        bin auto[4]                          15          1          -    Covered
        bin auto[5]                          24          1          -    Covered
        bin auto[6]                          19          1          -    Covered
        bin auto[7]                          13          1          -    Covered
        bin auto[8]                          18          1          -    Covered
        bin auto[9]                          20          1          -    Covered
        bin auto[10]                         18          1          -    Covered
        bin auto[11]                         18          1          -    Covered
        bin auto[12]                         22          1          -    Covered
        bin auto[13]                         15          1          -    Covered
        bin auto[14]                         18          1          -    Covered
        bin auto[15]                         34          1          -    Covered
    Coverpoint count_down_c2            100.00%        100          -    Covered
        covered/total bins:                   1          1          -
        missing/total bins:                   0          1          -
        % Hit:                          100.00%        100          -
        bin trans_underflow                   5          1          -    Covered

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1

ASSERTION RESULTS:
---------------------------------------------------------------------
Name                File(Line)                   Failure      Pass
                                                 Count        Count
---------------------------------------------------------------------
/counter_tb/#ublk#95084642#39/immed__41
                    counter_tb.svh(41)                0          1

Total Coverage By Instance (filtered view): 85.76%
```

2) ALSU is a logic unit that can perform logical, arithmetic, and shift operations on input ports

- Input ports A and B have various operations that can take place depending on the value of the opcode.
- Each input bit except for the clk and rst will be sampled at the rising edge before any processing so a D-FF is expected for each input bit at the design entry.
- The output of the ALSU is registered and is available at the rising edge of the clock.



Question 2

I. Verification plan

Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
ALSU_1	When reset is asserted, the outputs out and leds should be zero.	Reset is asserted randomly during simulation, with constraints that drive it to be off 90% of the time.	Check all transitions of out and leds under reset condition.	A checker in the testbench ensures that out and leds are zero when reset is high.
ALSU_2	When reset is deasserted, the output should match the operation defined by opcode with inputs A and B.	Randomized values of A, B, and opcode under constraints, with reset deasserted 90% of the time.	Cover all values and transitions of A, B, opcode, out, and leds.	A checker in the testbench ensures that out matches the expected operation result.
ALSU_3	Ensure that the ALSU produces correct results for all operations (OR, XOR, ADD, MULT, SHIFT, ROTATE).	Fully randomize inputs A, B, opcode, and control signals for multiple cycles.	Cover all operations based on opcodes and ensure transition coverage.	A checker in the testbench ensures the correctness of each operation based on opcode.
ALSU_4	Ensure correct handling of bypass_A and bypass_B, which should bypass inputs A and B respectively.	Randomize bypass_A, bypass_B signals and check if the inputs are bypassed or processed according to the control signals.	Functional coverage of bypass scenarios with A and B.	A checker ensures that the output behaves correctly when either or both bypass signals are asserted.
ALSU_5	Ensure correct handling of reduction operations on A and B using red_op_A and red_op_B.	Randomly toggle reduction operation signals red_op_A and red_op_B and monitor the effect on the output.	Cover all possible combinations of reduction operations for inputs A and B.	A checker ensures that reduction operations produce the correct results.
ALSU_6	Test ALSU functionality under high toggle rates of clock and inputs (A, B, opcode, etc.).	Rapidly toggle clock, inputs, and control signals to stress-test the ALSU's performance under frequent changes.	Functional coverage for frequent changes in inputs and impact on output signals.	A checker ensures the design works reliably under rapid input and clock changes.
ALSU_7	Ensure the correct behavior of the full adder when FULL_ADDER parameter is enabled/disabled.	Enable and disable the full adder randomly during tests	Ensure coverage of both FULL_ADDER on and off scenarios.	A checker ensures that carry is correctly handled when FULL_ADDER is enabled, and ignored otherwise.
ALSU_8	Verify the ALSU's performance in handling overflow and underflow conditions in arithmetic operations.	Randomize A and B values to stress-test for overflow and underflow in addition and multiplication operations.	Cover all corner cases of overflow and underflow for arithmetic operations.	A checker ensures that overflow/underflow conditions are handled correctly without errors.

II. RTL design

```
module ALSU(A, B, cin, serial_in, red_op_A, red_op_B, opcode, bypass_A, bypass_B, clk, rst, direction, leds, out);
parameter INPUT_PRIORITY = "A";
parameter FULL_ADDER = "ON";
input clk, rst, red_op_A, red_op_B, bypass_A, bypass_B, direction, serial_in;
input [2:0] opcode;
input signed cin;
input signed [2:0] A, B;
output reg [15:0] leds;
output reg signed [5:0] out;

reg red_op_A_reg, red_op_B_reg, bypass_A_reg, bypass_B_reg, direction_reg, serial_in_reg;
reg [2:0] opcode_reg;
reg signed [2:0] A_reg, B_reg;
reg signed cin_reg ;

wire invalid_red_op, invalid_opcode, invalid;

//Invalid handling
assign invalid_red_op = (red_op_A_reg | red_op_B_reg) & (opcode_reg[1] | opcode_reg[2]);
assign invalid_opcode = opcode_reg[1] & opcode_reg[2];
assign invalid = invalid_red_op | invalid_opcode;

//Registering input signals
always @(posedge clk or posedge rst) begin
  if(rst) begin
     cin_reg <= 0;
     red_op_B_reg <= 0;
     red_op_A_reg <= 0;
     bypass_B_reg <= 0;
     bypass_A_reg <= 0;
     direction_reg <= 0;
     serial_in_reg <= 0;
     opcode_reg <= 0;
     A_reg <= 0;
     B_reg <= 0;
  end else begin
     cin_reg <= cin;
     red_op_B_reg <= red_op_B;
     red_op_A_reg <= red_op_A;
     bypass_B_reg <= bypass_B;
     bypass_A_reg <= bypass_A;
     direction_reg <= direction;
     serial_in_reg <= serial_in;
     opcode_reg <= opcode;
     A_reg <= A;
     B_reg <= B;
  end
end

//leds output blinking
always @(posedge clk or posedge rst) begin
  if(rst) begin
     leds <= 0;
  end else begin
      if (invalid)
        leds <= ~leds;
      else
        leds <= 0;
```

```
    end
  end

  //ALSU output processing
  always @(posedge clk or posedge rst) begin
    if(rst) begin
      out <= 0;
    end
    else begin
      if (invalid)
          out <= 0;
      else if (bypass_A_reg && bypass_B_reg)
        out <= (INPUT_PRIORITY == "A")? A_reg: B_reg;
      else if (bypass_A_reg)
        out <= A_reg;
      else if (bypass_B_reg)
        out <= B_reg;
      else begin
          case (opcode)
    3'h0: begin
      if (red_op_A_reg && red_op_B_reg)
        out <= (INPUT_PRIORITY == "A") ? |A_reg : |B_reg;
      else if (red_op_A_reg)
        out <= |A_reg;
      else if (red_op_B_reg)
        out <= |B_reg;
      else
        out <= A_reg | B_reg;
    end
    3'h1: begin
      if (red_op_A_reg && red_op_B_reg)
        out <= (INPUT_PRIORITY == "A") ? ^A_reg : ^B_reg;
      else if (red_op_A_reg)
        out <= ^A_reg;
      else if (red_op_B_reg)
        out <= ^B_reg;
      else
        out <= A_reg ^ B_reg;
    end
            3'h2: begin
               if (FULL_ADDER == "ON")
               out <= A_reg + B_reg + cin_reg;
               else out <= A_reg + B_reg;
            end
            3'h3: out <= A_reg * B_reg;
            3'h4: begin
              if (direction_reg)
                out <= {out[4:0], serial_in_reg};
              else
                out <= {serial_in_reg, out[5:1]};
            end
            3'h5: begin
              if (direction_reg)
                out <= {out[4:0], out[5]};
              else
                out <= {out[0], out[5:1]};
            end
             default : out <= 0;

          endcase
      end
    end
  end

  endmodule
```

III. Golden Model

```
module ALSU_GM(A, B, cin, serial_in, red_op_A, red_op_B, opcode, bypass_A, bypass_B, clk, rst, direction, leds_expected, out_expected);
input clk, rst, red_op_A, red_op_B, bypass_A, bypass_B, direction, serial_in;
input [2: 0] opcode;
input signed cin;
input signed [2: 0] A, B;
output reg [15: 0] leds_expected;
output reg signed [5: 0] out_expected;
reg red_op_A_reg, red_op_B_reg, bypass_A_reg, bypass_B_reg, direction_reg, serial_in_reg;
reg [2: 0] opcode_reg;
reg signed [2: 0] A_reg,B_reg;
reg signed cin_reg;
wire invalid_red_op, invalid_opcode, invalid;
assign invalid_red_op = (red_op_A_reg | red_op_B_reg) & (opcode_reg[1] | opcode_reg[2]);
assign invalid_opcode = opcode_reg[1] & opcode_reg[2];
assign invalid = invalid_red_op | invalid_opcode;
always @(posedge clk or posedge rst) begin
if(rst) begin
cin_reg <= 0;
red_op_B_reg <= 0;
red_op_A_reg <= 0;
bypass_B_reg <= 0;
bypass_A_reg <= 0;
direction_reg <= 0;
serial_in_reg <= 0;
opcode_reg <= 0;
A_reg <= 0;
B_reg <= 0;
end else begin
cin_reg <= cin;
red_op_B_reg <= red_op_B;
red_op_A_reg <= red_op_A;
bypass_B_reg <= bypass_B;
bypass_A_reg <= bypass_A;
direction_reg <= direction;
serial_in_reg <= serial_in;
opcode_reg <= opcode;
A_reg <= A;
B_reg <= B;
end
end
always @(posedge clk or posedge rst) begin
if(rst) begin
leds_expected <= 0;
end else begin
if (invalid)
leds_expected <= ~leds_expected;
else
leds_expected <= 0;
end
end
always @(posedge clk or posedge rst) begin
if(rst) begin
out_expected <= 0;
end
else begin
if (invalid)
out_expected <= 0;
else if (bypass_A_reg && bypass_B_reg)
out_expected <= A_reg ;
else if (bypass_A_reg)
```

```
61    out_expected <= A_reg ;
62    else if (bypass_B_reg)
63    out_expected <= B_reg ;
64    else begin
65    case (opcode)
66    3'h0:
67    if (red_op_A_reg && red_op_B_reg)
68    out_expected = |A_reg;
69    else if (red_op_A_reg)
70    out_expected <= |A_reg;
71    else if (red_op_B_reg)
72    out_expected <= |B_reg;
73    else
74    out_expected <= A_reg | B_reg;
75    3'h1:
76    if (red_op_A_reg && red_op_B_reg)
77    out_expected <= ^A_reg;
78    else if (red_op_A_reg)
79    out_expected <= ^A_reg;
80    else if (red_op_B_reg)
81    out_expected <= ^B_reg;
82    else
83    out_expected <= A_reg ^ B_reg;
84    3'h2: out_expected <= A_reg + B_reg + cin_reg ;
85    3'h3: out_expected <= A_reg * B_reg;
86    3'h4:
87    if (direction_reg)
88    out_expected <= {out_expected[4: 0], serial_in_reg};
89    else
90    out_expected <= {serial_in_reg, out_expected[5: 1]};
91    3'h5:
92    if (direction_reg)
93    out_expected <= {out_expected[4: 0], out_expected[5]};
94    else
95    out_expected <= {out_expected[0], out_expected[5: 1]};
96    default: out_expected <= 0 ;
97    endcase
98    end
99    end
100   end
101   endmodule
102
```

IV. Package

```
package pck;
typedef enum bit [2: 0] {OR, XOR, ADD, MULT, SHIFT,ROTATE,INVALID_6,INVALID_7} opcode_e ;
typedef enum bit [2: 0] {Or, Xor, Add, Mult, Shift,Rotate} opcode_valid_e ;

localparam MAXPOS = 3 ;
localparam MAXNEG = -4 ;
localparam ZERO = 0 ;
class Z;
rand bit rst_c , red_op_A_c , red_op_B_c , cin_c , serial_in_c , direction_c , bypass_A_c , bypass_B_c ;
rand bit signed [2: 0] A_c , B_c ;

rand opcode_e opcode_c;
rand opcode_valid_e array[6];
constraint c_rst {rst_c dist {0:/90 , 1:/10};}
constraint c_input_A {
  if (opcode_c == ADD || opcode_c == MULT) {
    A_c dist {
      MAXPOS / 25,
      MAXNEG / 25,
      ZERO / 25,
      {3'b001, 3'b010, 3'b101, 3'b110, 3'b111} / 25
    };} else if ((opcode_c == OR || opcode_c == XOR) && red_op_A_c == 1) {B_c == 0;
    A_c dist {
      {3'b001, 3'b010, 3'b100} := 80,
      {3'b000, 3'b011, 3'b101, 3'b110, 3'b111} := 20
    };} else {
    A_c inside {[MAXNEG: MAXPOS]};}}
constraint c_input_B {
  if (opcode_c == ADD || opcode_c == MULT) {
    B_c dist {
      MAXPOS / 25,
      MAXNEG / 25,
      ZERO / 25,
      {3'b001, 3'b010, 3'b101, 3'b110, 3'b111} / 25};
  } else if ((opcode_c == OR || opcode_c == XOR) && red_op_A_c == 1) {
    A_c == 0;
    B_c dist {
      {3'b001, 3'b010, 3'b100} := 80,
      {3'b000, 3'b011, 3'b101, 3'b110, 3'b111} := 20 };}
      else {B_c inside {[MAXNEG: MAXPOS]}; }}
constraint c_opcode { opcode_c dist { [0: 5]:/90 ,[6: 7]:/10 }; };
constraint c_bypass_signals {bypass_A_c dist { 0:/90 , 1:/10 };bypass_B_c dist { 0:/90 , 1:/10 };};
 // Additional constraints to ensure all branches are covered
  constraint all_branches {
    // Test all combinations for bypass signals and red_op signals
    if (opcode_c == ADD || opcode_c == MULT) {
      red_op_A_c dist {0:/50, 1:/50};
      red_op_B_c dist {0:/50, 1:/50};}
    // For OR and XOR operations
    if (opcode_c == OR || opcode_c == XOR) {
```

```
        // Test if red_op_A_c is true
        if (red_op_A_c == 1) {
          A_c dist { {3'b001, 3'b010, 3'b100} := 80, {3'b000, 3'b011, 3'b101, 3'b110, 3'b111} := 20 };B_c == 0;}
        // Test if red_op_B_c is true
        if (red_op_B_c == 1) {B_c dist { {3'b001, 3'b010, 3'b100} := 80, {3'b000, 3'b011, 3'b101, 3'b110, 3'b111} := 20 }; A_c == 0;}}}
constraint c_array {foreach (array[i]) {foreach (array[j]) {if (i != j) {array[i] != array[j];}}}}

covergroup cg ;
A_cp: coverpoint A_c
{
bins data_0 = {ZERO};
bins data_max = {MAXPOS};
bins data_min = {MAXNEG};
bins data_walkingones = {3'b001,3'b010,3'b100};
bins data_default = default ;
}
B_cp: coverpoint B_c
{
bins data_0 = {ZERO};
bins data_max = {MAXPOS};
bins data_min = {MAXNEG};
bins data_walkingones = {3'b001,3'b010,3'b100};
bins data_default = default ;

}
ALU_cp: coverpoint opcode_c
{
bins Bins_shift[] = {SHIFT, ROTATE};
bins Bins_arith[] = {ADD, MULT};
bins Bins_bitwise[] = {OR, XOR};
illegal_bins Bins_invalid = {INVALID_6,INVALID_7};
bins Bins_trans = (OR => XOR => ADD => MULT => SHIFT => ROTATE);
}

endgroup
function new();
cg = new();
endfunction
endclass
endpackage
```

V. Test Bench

```
import pck::*;

module ALSU_TB;
    // Parameters declaration
    parameter INPUT_PRIORITY = "A";
    parameter FULL_ADDER = "ON";

    // Inputs/Outputs declaration
    logic clk, rst, cin, red_op_A, red_op_B, bypass_A, bypass_B, direction, serial_in;
    logic [2:0] opcode;
    logic signed [2:0] A, B;
    logic [15:0] leds;
    logic signed [5:0] out;
    logic signed [5:0] out_expected;
    logic [15:0] leds_expected;
    opcode_valid_e opvalid[6];

    // Create object from the class
    Z obj;
    int error_count, correct_count;

    // Module instantiation for Design and Golden Model
    ALSU #(.INPUT_PRIORITY(INPUT_PRIORITY), .FULL_ADDER(FULL_ADDER)) DUT (.*);
    ALSU_GM DUT2(.*);

    // Clock generation
    initial begin
        clk = 0;
        forever #25 clk = ~clk;
    end

    // Main program block
    initial begin
        obj = new();
        error_count = 0;
        correct_count = 0;

        // First part with loop
        obj.c_array.constraint_mode(0);
        for(int i = 0; i < 10000; i++) begin
            @(negedge clk);
            assert(obj.randomize());
            rst = obj.rst_c;
            cin = obj.cin_c;
            red_op_A = obj.red_op_A_c;
            red_op_B = obj.red_op_B_c;
            bypass_A = obj.bypass_A_c;
            bypass_B = obj.bypass_B_c;
            direction = obj.direction_c;
            serial_in = obj.serial_in_c;
            opcode = obj.opcode_c;
            A = obj.A_c;
            B = obj.B_c;
            sample_data();
            golden_model();
        end
```

```
    // Second part with loop
    obj.constraint_mode(0);
    obj.c_array.constraint_mode(1);

    // Force reduction operation and bypass to zeros
    red_op_A = 0;
    red_op_B = 0;
    bypass_A = 0;
    bypass_B = 0;

    for(int i = 0; i < 10000; i++) begin
        @(negedge clk);
        assert(obj.randomize());
        rst = obj.rst_c;
        cin = obj.cin_c;
        red_op_A = obj.red_op_A_c;
        red_op_B = obj.red_op_B_c;
        bypass_A = obj.bypass_A_c;
        bypass_B = obj.bypass_B_c;
        direction = obj.direction_c;
        serial_in = obj.serial_in_c;
        A = obj.A_c;
        B = obj.B_c;

        for(int j = 0; j < 6; j++) begin
            opvalid[j] = obj.array[j];
            opcode = opvalid[j];
            #25 sample_data();
        end
        sample_data();
        golden_model();
    end
// Direct case to hit the transition bin
    // Initialize coverage collection
    obj.cg.start();

    // Test transitions from opcode 0 to 5
    obj.opcode_c = opcode_e'(3'b000);  // OR
    opcode = 3'b000;
    @(negedge clk);
    #25;  // Wait for half a clock period
    obj.cg.sample();

    obj.opcode_c = opcode_e'(3'b001);  // XOR
    opcode = 3'b001;
    @(negedge clk);
    #25;
    obj.cg.sample();

    obj.opcode_c = opcode_e'(3'b010);  // ADD
    opcode = 3'b010;
    @(negedge clk);
    #25;
    obj.cg.sample();
```

```
        obj.cg.sample();

        obj.opcode_c = opcode_e'(3'b011);  // MULT
        opcode = 3'b011;
        @(negedge clk);
        #25;
        obj.cg.sample();

        obj.opcode_c = opcode_e'(3'b100);  // SHIFT
        opcode = 3'b100;
        @(negedge clk);
        #25;
        obj.cg.sample();

        obj.opcode_c = opcode_e'(3'b101);  // ROTATE
        opcode = 3'b101;
        @(negedge clk);
        #25;
        obj.cg.sample();
        $display("correct counter = %0d , error counter = %0d", correct_count, error_count);
        $stop;
    end

    task sample_data();
        if (rst == 1 || bypass_A == 1 || bypass_B == 1) begin
            obj.cg.stop();
        end else begin
            obj.cg.start();
            obj.cg.sample();
        end
    endtask

    task golden_model();
        @(negedge clk);
        if (out != out_expected || leds != leds_expected) begin
            $display("Incorrect result!! A=%0d, B=%0d, opcode=%0b, out=%0d, out_expected=%0d, leds=%0b, leds_expected=%0b",A, B, opcode, out, out_expected, leds, leds_expected);
            error_count++;
        end else begin
            correct_count++;
        end
    endtask

endmodule
```

VI. Print

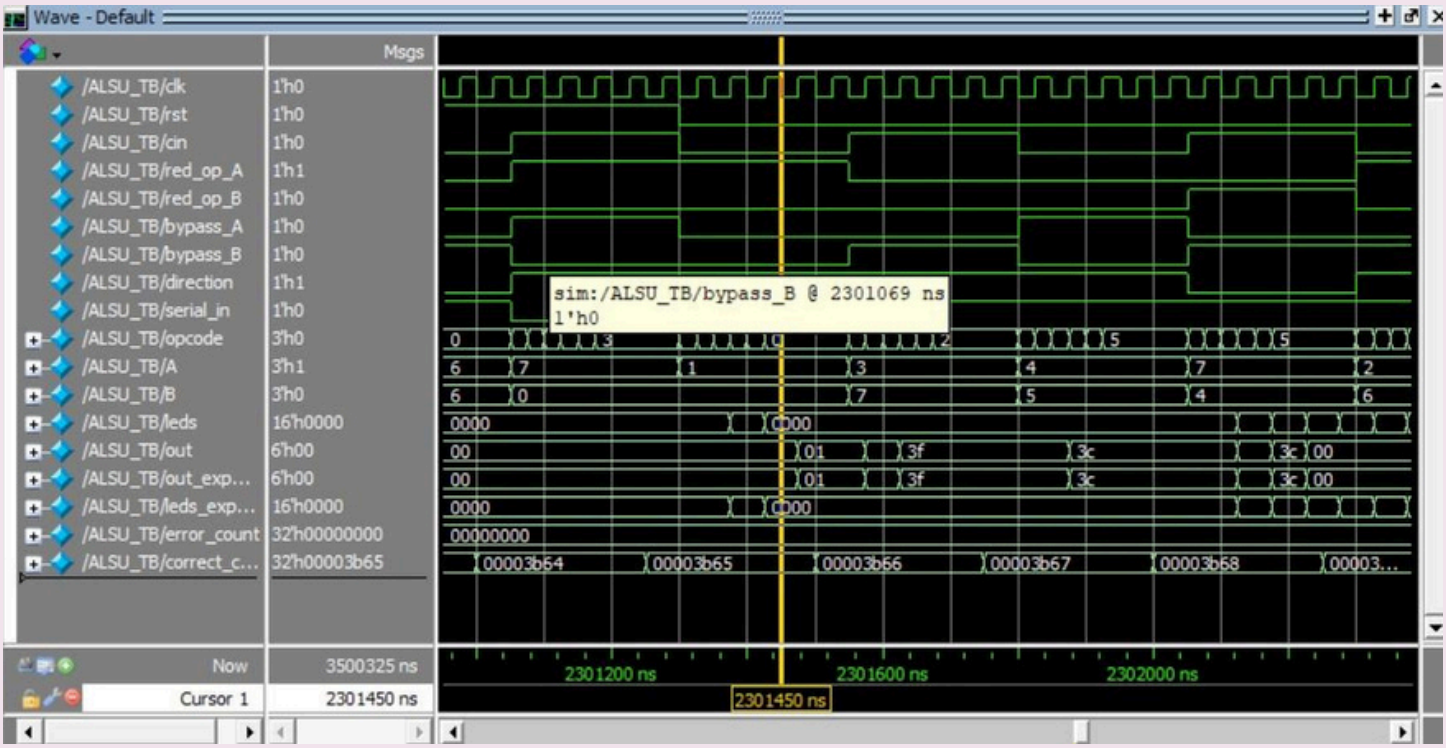
```
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1820.
   Time: 2303075 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1821.
   Time: 2303100 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1822.
   Time: 2303125 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1823.
   Time: 2303150 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1824.
   Time: 2303175 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1825.
   Time: 2303200 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1826.
   Time: 2303200 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1827.
   Time: 2307575 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1828.
   Time: 2307600 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1829.
   Time: 2307625 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1830.
   Time: 2307650 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1831.
   Time: 2307675 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1832.
   Time: 2307700 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1833.
   Time: 2307700 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1834.
   Time: 2315325 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1835.
   Time: 2315350 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1836.
   Time: 2315375 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1837.
   Time: 2315400 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1838.
   Time: 2315425 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1839.
   Time: 2315450 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1840.
   Time: 2315450 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1841.
   Time: 2364325 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1842.
   Time: 2364350 ns  Iteration: 0  Instance: /ALSU_TB
```

```
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1797.
   Time: 2280950 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1798.
   Time: 2280950 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1799.
   Time: 2282575 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1800.
   Time: 2282600 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1801.
   Time: 2282625 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1802.
   Time: 2282650 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1803.
   Time: 2282675 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1804.
   Time: 2282700 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1805.
   Time: 2282700 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1806.
   Time: 2297325 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1807.
   Time: 2297350 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1808.
   Time: 2297375 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1809.
   Time: 2297400 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1810.
   Time: 2297425 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1811.
   Time: 2297450 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_6. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1812.
   Time: 2297450 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1813.
   Time: 2301325 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1814.
   Time: 2301350 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1815.
   Time: 2301375 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1816.
   Time: 2301400 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1817.
   Time: 2301425 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1818.
   Time: 2301450 ns  Iteration: 0  Instance: /ALSU_TB
** Error: (vsim-8565) Illegal state bin was hit at value=INVALID_7. The bin counter for the illegal bin '\/pck::Z::cg .ALU_cp.Bins_invalid' is 1819.
```

VII. Do file

```
1  vlib work
2  vlog ALSU.v Package.sv ALSU_GM.svh ALSU_TB.sv  +cover -covercells
3  vsim -voptargs=+acc work.ALSU_TB -cover
4  add wave *
5  coverage save ALSU_TB.ucdb -onexit -du work.ALSU
6  run -all
```

VIII. Waveform



IX. Functional Coverage Report

```
Coverage Report by instance with details

================================================================================
=== Instance: /pck
=== Design Unit: work.pck
================================================================================

Covergroup Coverage:
    Covergroups                 1        na        na   100.00%
        Coverpoints/Crosses     3        na        na        na
            Covergroup Bins    15        15         0   100.00%
--------------------------------------------------------------------------------------------------------
Covergroup                                          Metric       Goal       Bins    Status

--------------------------------------------------------------------------------------------------------
 TYPE /pck/Z/cg                                    100.00%        100          -    Covered
    covered/total bins:                                 15         15          -
    missing/total bins:                                  0         15          -
    % Hit:                                          100.00%        100          -
    Coverpoint A_cp                                 100.00%        100          -    Covered
        covered/total bins:                              4          4          -
        missing/total bins:                              0          4          -
        % Hit:                                      100.00%        100          -
    Coverpoint B_cp                                 100.00%        100          -    Covered
        covered/total bins:                              4          4          -
        missing/total bins:                              0          4          -
        % Hit:                                      100.00%        100          -
    Coverpoint ALU_cp                               100.00%        100          -    Covered
        covered/total bins:                              7          7          -
        missing/total bins:                              0          7          -
        % Hit:                                      100.00%        100          -
 Covergroup instance \/pck::Z::cg                   100.00%        100          -    Covered
    covered/total bins:                                 15         15          -
    missing/total bins:                                  0         15          -
    % Hit:                                          100.00%        100          -
    Coverpoint A_cp                                 100.00%        100          -    Covered
        covered/total bins:                              4          4          -
        missing/total bins:                              0          4          -
        % Hit:                                      100.00%        100          -
        bin data_0                                     3913          1          -    Covered
        bin data_max                                   1928          1          -    Covered
        bin data_min                                   1761          1          -    Covered
        bin data_walkingones                           3529          1          -    Covered
        default bin data_default                       5231                     -    Occurred
    Coverpoint B_cp                                 100.00%        100          -    Covered
        covered/total bins:                              4          4          -
        missing/total bins:                              0          4          -
        % Hit:                                      100.00%        100          -
        bin data_0                                     3964          1          -    Covered
        bin data_max                                   1739          1          -    Covered
        bin data_min                                   1706          1          -    Covered
        bin data_walkingones                           3321          1          -    Covered
        default bin data_default                       5632                     -    Occurred
    Coverpoint ALU_cp                               100.00%        100          -    Covered
        covered/total bins:                              7          7          -
        missing/total bins:                              0          7          -
        % Hit:                                      100.00%        100          -
        illegal_bin Bins_invalid                       2939                     -    Occurred
        bin Bins_shift[SHIFT]                          2238          1          -    Covered
        bin Bins_shift[ROTATE]                         2144          1          -    Covered
        bin Bins_arith[ADD]                            2209          1          -    Covered
        bin Bins_arith[MULT]                           2390          1          -    Covered
        bin Bins_bitwise[OR]                           2240          1          -    Covered
        bin Bins_bitwise[XOR]                          2202          1          -    Covered
        bin Bins_trans                                    1          1          -    Covered

COVERGROUP COVERAGE:
--------------------------------------------------------------------------------------------------------
Covergroup                                          Metric       Goal       Bins    Status

--------------------------------------------------------------------------------------------------------
 TYPE /pck/Z/cg                                    100.00%        100          -    Covered
    covered/total bins:                                 15         15          -
    missing/total bins:                                  0         15          -
    % Hit:                                          100.00%        100          -
    Coverpoint A_cp                                 100.00%        100          -    Covered
        covered/total bins:                              4          4          -
        missing/total bins:                              0          4          -
        % Hit:                                      100.00%        100          -
    Coverpoint B_cp                                 100.00%        100          -    Covered
        covered/total bins:                              4          4          -
        missing/total bins:                              0          4          -
        % Hit:                                      100.00%        100          -
    Coverpoint ALU_cp                               100.00%        100          -    Covered
        covered/total bins:                              7          7          -
        missing/total bins:                              0          7          -
        % Hit:                                      100.00%        100          -
```

```
        missing/total bins:                    0          7          -
        % Hit:                            100.00%        100          -
  Covergroup instance \/pck::Z::cg        100.00%        100          -    Covered
    covered/total bins:                        15         15          -
    missing/total bins:                         0         15          -
    % Hit:                                100.00%        100          -
    Coverpoint A_cp                       100.00%        100          -    Covered
        covered/total bins:                     4          4          -
        missing/total bins:                     0          4          -
        % Hit:                            100.00%        100          -
        bin data_0                           3913          1          -    Covered
        bin data_max                         1928          1          -    Covered
        bin data_min                         1761          1          -    Covered
        bin data_walkingones                 3529          1          -    Covered
        default bin data_default             5231                     -    Occurred
    Coverpoint B_cp                       100.00%        100          -    Covered
        covered/total bins:                     4          4          -
        missing/total bins:                     0          4          -
        % Hit:                            100.00%        100          -
        bin data_0                           3964          1          -    Covered
        bin data_max                         1739          1          -    Covered
        bin data_min                         1706          1          -    Covered
        bin data_walkingones                 3321          1          -    Covered
        default bin data_default             5632                     -    Occurred
    Coverpoint ALU_cp                     100.00%        100          -    Covered
        covered/total bins:                     7          7          -
        missing/total bins:                     0          7          -
        % Hit:                            100.00%        100          -
        illegal_bin Bins_invalid             2939                     -    Occurred
        bin Bins_shift[SHIFT]                2238          1          -    Covered
        bin Bins_shift[ROTATE]               2144          1          -    Covered
        bin Bins_arith[ADD]                  2209          1          -    Covered
        bin Bins_arith[MULT]                 2390          1          -    Covered
        bin Bins_bitwise[OR]                 2240          1          -    Covered
        bin Bins_bitwise[XOR]                2202          1          -    Covered
        bin Bins_trans                          1          1          -    Covered

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1

Total Coverage By Instance (filtered view): 100.00%
```

X. Coverage Report

```
Coverage Report by instance with details

================================================================================
=== Instance: /\ALSU_TB#DUT
=== Design Unit: work.ALSU
================================================================================
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        32        32         0   100.00%

================================Branch Details================================

Branch Coverage for instance /\ALSU_TB#DUT

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File ALSU.v
------------------------------------IF Branch------------------------------------
    25                                     57411     Count coming in to IF
    25              1                      19406
    36              1                      38005
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    52                                     73450     Count coming in to IF
    52              1                      30396
    54              1                      43054
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    55                                     43054     Count coming in to IF
    55              1                      20642
    57              1                      22412
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    64                                     56742     Count coming in to IF
    64              1                      19394
    67              1                      37348
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    68                                     37348     Count coming in to IF
    68              1                      17678
    70              1                       2056
    72              1                       2789
    74              1                       2766
    76              1                      12059
Branch totals: 5 hits of 5 branches = 100.00%

----------------------------------CASE Branch----------------------------------
    77                                     12059     Count coming in to CASE
    78              1                       2595
    88              1                       2544
    98              1                       1592
    103             1                       1589
    104             1                       1651
    110             1                       1696
    116             1                        392
Branch totals: 7 hits of 7 branches = 100.00%

------------------------------------IF Branch------------------------------------
    79                                      2595     Count coming in to IF
    79              1                        109
    81              1                        106
    83              1                        112
    85              1                       2268
Branch totals: 4 hits of 4 branches = 100.00%

------------------------------------IF Branch------------------------------------
    89                                      2544     Count coming in to IF
    89              1                        106
    91              1                         94
    93              1                         77
    95              1                       2267
Branch totals: 4 hits of 4 branches = 100.00%

------------------------------------IF Branch------------------------------------
    105                                     1651     Count coming in to IF
    105             1                        549
    107             1                       1102
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    111                                     1696     Count coming in to IF
    111             1                        573
    113             1                       1123
Branch totals: 2 hits of 2 branches = 100.00%
```

```
Condition Coverage:
    Enabled Coverage              Bins   Covered    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Conditions                       6         6         0   100.00%

================================Condition Details================================

Condition Coverage for instance /\ALSU_TB#DUT  --

  File ALSU.v
----------------Focused Condition View-------------------
Line       70 Item    1  (bypass_A_reg && bypass_B_reg)
Condition totals: 2 of 2 input terms covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
    ----------  --------  ----------------------  --------------
  bypass_A_reg         Y
  bypass_B_reg         Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  bypass_A_reg_0        -
  Row   2:          1  bypass_A_reg_1        bypass_B_reg
  Row   3:          1  bypass_B_reg_0        bypass_A_reg
  Row   4:          1  bypass_B_reg_1        bypass_A_reg

----------------Focused Condition View-------------------
Line       79 Item    1  (red_op_A_reg && red_op_B_reg)
Condition totals: 2 of 2 input terms covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
    ----------  --------  ----------------------  --------------
  red_op_A_reg         Y
  red_op_B_reg         Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  red_op_A_reg_0        -
  Row   2:          1  red_op_A_reg_1        red_op_B_reg
  Row   3:          1  red_op_B_reg_0        red_op_A_reg
  Row   4:          1  red_op_B_reg_1        red_op_A_reg

----------------Focused Condition View-------------------
Line       89 Item    1  (red_op_A_reg && red_op_B_reg)
Condition totals: 2 of 2 input terms covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
    ----------  --------  ----------------------  --------------
  red_op_A_reg         Y
  red_op_B_reg         Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  red_op_A_reg_0        -
  Row   2:          1  red_op_A_reg_1        red_op_B_reg
  Row   3:          1  red_op_B_reg_0        red_op_A_reg
  Row   4:          1  red_op_B_reg_1        red_op_A_reg


Expression Coverage:
    Enabled Coverage              Bins   Covered    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Expressions                      8         8         0   100.00%

================================Expression Details================================

Expression Coverage for instance /\ALSU_TB#DUT  --

  File ALSU.v
----------------Focused Expression View-----------------
Line       19 Item    1  ((red_op_A_reg | red_op_B_reg) & (opcode_reg[1] | opcode_reg[2]))
Expression totals: 4 of 4 input terms covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
    ----------  --------  ----------------------  --------------
  red_op_A_reg         Y
  red_op_B_reg         Y
 opcode_reg[1]         Y
 opcode_reg[2]         Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  red_op_A_reg_0        ((opcode_reg[1] | opcode_reg[2]) && ~red_op_B_reg)
  Row   2:          1  red_op_A_reg_1        ((opcode_reg[1] | opcode_reg[2]) && ~red_op_B_reg)
  Row   3:          1  red_op_B_reg_0        ((opcode_reg[1] | opcode_reg[2]) && ~red_op_A_reg)
  Row   4:          1  red_op_B_reg_1        ((opcode_reg[1] | opcode_reg[2]) && ~red_op_A_reg)
  Row   5:          1  opcode_reg[1]_0       ((red_op_A_reg | red_op_B_reg) && ~opcode_reg[2])
  Row   6:          1  opcode_reg[1]_1       ((red_op_A_reg | red_op_B_reg) && ~opcode_reg[2])
  Row   7:          1  opcode_reg[2]_0       ((red_op_A_reg | red_op_B_reg) && ~opcode_reg[1])
  Row   8:          1  opcode_reg[2]_1       ((red_op_A_reg | red_op_B_reg) && ~opcode_reg[1])
```

```
Condition Coverage:
    Enabled Coverage              Bins   Covered    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Conditions                       6         6         0   100.00%

================================Condition Details================================

Condition Coverage for instance /\ALSU_TB#DUT  --

  File ALSU.v
----------------Focused Condition View-------------------
Line       70 Item    1  (bypass_A_reg && bypass_B_reg)
Condition totals: 2 of 2 input terms covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
    ----------  --------  ----------------------  --------------
  bypass_A_reg         Y
  bypass_B_reg         Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  bypass_A_reg_0        -
  Row   2:          1  bypass_A_reg_1        bypass_B_reg
  Row   3:          1  bypass_B_reg_0        bypass_A_reg
  Row   4:          1  bypass_B_reg_1        bypass_A_reg

----------------Focused Condition View-------------------
Line       79 Item    1  (red_op_A_reg && red_op_B_reg)
Condition totals: 2 of 2 input terms covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
    ----------  --------  ----------------------  --------------
  red_op_A_reg         Y
  red_op_B_reg         Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  red_op_A_reg_0        -
  Row   2:          1  red_op_A_reg_1        red_op_B_reg
  Row   3:          1  red_op_B_reg_0        red_op_A_reg
  Row   4:          1  red_op_B_reg_1        red_op_A_reg

----------------Focused Condition View-------------------
Line       89 Item    1  (red_op_A_reg && red_op_B_reg)
Condition totals: 2 of 2 input terms covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
    ----------  --------  ----------------------  --------------
  red_op_A_reg         Y
  red_op_B_reg         Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  red_op_A_reg_0        -
  Row   2:          1  red_op_A_reg_1        red_op_B_reg
  Row   3:          1  red_op_B_reg_0        red_op_A_reg
  Row   4:          1  red_op_B_reg_1        red_op_A_reg


Expression Coverage:
    Enabled Coverage              Bins   Covered    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Expressions                      8         8         0   100.00%

================================Expression Details================================

Expression Coverage for instance /\ALSU_TB#DUT  --

  File ALSU.v
----------------Focused Expression View-----------------
Line       19 Item    1  ((red_op_A_reg | red_op_B_reg) & (opcode_reg[1] | opcode_reg[2]))
Expression totals: 4 of 4 input terms covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
    ----------  --------  ----------------------  --------------
  red_op_A_reg         Y
  red_op_B_reg         Y
  opcode_reg[1]        Y
  opcode_reg[2]        Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  red_op_A_reg_0        ((opcode_reg[1] | opcode_reg[2]) && ~red_op_B_reg)
  Row   2:          1  red_op_A_reg_1        ((opcode_reg[1] | opcode_reg[2]) && ~red_op_B_reg)
  Row   3:          1  red_op_B_reg_0        ((opcode_reg[1] | opcode_reg[2]) && ~red_op_A_reg)
  Row   4:          1  red_op_B_reg_1        ((opcode_reg[1] | opcode_reg[2]) && ~red_op_A_reg)
  Row   5:          1  opcode_reg[1]_0       ((red_op_A_reg | red_op_B_reg) && ~opcode_reg[2])
  Row   6:          1  opcode_reg[1]_1       ((red_op_A_reg | red_op_B_reg) && ~opcode_reg[2])
  Row   7:          1  opcode_reg[2]_0       ((red_op_A_reg | red_op_B_reg) && ~opcode_reg[1])
  Row   8:          1  opcode_reg[2]_1       ((red_op_A_reg | red_op_B_reg) && ~opcode_reg[1])
```

```
----------------Focused Expression View-----------------
Line       20 Item    1  (opcode_reg[1] & opcode_reg[2])
Expression totals: 2 of 2 input terms covered = 100.00%

     Input Term   Covered  Reason for no coverage   Hint
     ----------  --------  ----------------------  --------------
  opcode_reg[1]         Y
  opcode_reg[2]         Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  opcode_reg[1]_0       opcode_reg[2]
  Row   2:          1  opcode_reg[1]_1       opcode_reg[2]
  Row   3:          1  opcode_reg[2]_0       opcode_reg[1]
  Row   4:          1  opcode_reg[2]_1       opcode_reg[1]

----------------Focused Expression View-----------------
Line       21 Item    1  (invalid_red_op | invalid_opcode)
Expression totals: 2 of 2 input terms covered = 100.00%

     Input Term   Covered  Reason for no coverage   Hint
     ----------  --------  ----------------------  --------------
  invalid_red_op        Y
  invalid_opcode        Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  invalid_red_op_0      ~invalid_opcode
  Row   2:          1  invalid_red_op_1      ~invalid_opcode
  Row   3:          1  invalid_opcode_0      ~invalid_red_op
  Row   4:          1  invalid_opcode_1      ~invalid_red_op


Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                      49        49         0   100.00%

================================Statement Details================================

Statement Coverage for instance /\ALSU_TB#DUT  --

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File ALSU.v
    19              1                      26320
    20              1                      24812
    21              1                      13629
    24              1                      57411
    26              1                      19406
    27              1                      19406
    28              1                      19406
    29              1                      19406
    30              1                      19406
    31              1                      19406
    32              1                      19406
    33              1                      19406
    34              1                      19406
    35              1                      19406
    37              1                      38005
    38              1                      38005
    39              1                      38005
    40              1                      38005
    41              1                      38005
    42              1                      38005
    43              1                      38005
    44              1                      38005
    45              1                      38005
    46              1                      38005
    51              1                      73450
    53              1                      30396
    56              1                      20642
    58              1                      22412
    63              1                      56742
    65              1                      19394
    69              1                      17678
    71              1                       2056
    73              1                       2789
    75              1                       2766
    80              1                        109
    82              1                        106
    84              1                        112
    86              1                       2268
    90              1                        106
    92              1                         94
    94              1                         77
    96              1                       2267
    100             1                       1592
    103             1                       1589
    106             1                        549
    108             1                       1102
    112             1                        573
    114             1                       1123
    116             1                        392
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
```

```
    Toggles                          120       120          0   100.00%

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

Toggle Coverage for instance /\ALSU_TB#DUT  --

                                          Node       1H->0L       0L->1H  "Coverage"
                                          ----------------------------------------
                                          A[0-2]            1            1      100.00
                                        A_reg[2-0]          1            1      100.00
                                          B[0-2]            1            1      100.00
                                        B_reg[2-0]          1            1      100.00
                                         bypass_A           1            1      100.00
                                       bypass_A_reg         1            1      100.00
                                         bypass_B           1            1      100.00
                                       bypass_B_reg         1            1      100.00
                                           cin              1            1      100.00
                                        cin_reg[1-0]        1            1      100.00
                                           clk              1            1      100.00
                                        direction           1            1      100.00
                                      direction_reg         1            1      100.00
                                         invalid            1            1      100.00
                                      invalid_opcode        1            1      100.00
                                      invalid_red_op        1            1      100.00
                                        leds[15-0]          1            1      100.00
                                       opcode[0-2]          1            1      100.00
                                     opcode_reg[2-0]        1            1      100.00
                                         out[5-0]           1            1      100.00
                                         red_op_A           1            1      100.00
                                       red_op_A_reg         1            1      100.00
                                         red_op_B           1            1      100.00
                                       red_op_B_reg         1            1      100.00
                                           rst              1            1      100.00
                                       serial_in            1            1      100.00
                                     serial_in_reg          1            1      100.00

Total Node Count      =          60
Toggled Node Count    =          60
Untoggled Node Count  =           0

Toggle Coverage       =     100.00% (120 of 120 bins)


Total Coverage By Instance (filtered view): 100.00%
```

Q3. The design to be tested is a synchronous single-port 8-bit x64K (512kBit) RAM. The RAM will read on the positive edge of the clock when input read =1 and write on the positive edge of the clock when input write = 1. Write enable signal has a higher priority than the read enable signal and both write and read data from the RAM is not allowed at the same time. Even parity will be calculated on data written to the RAM and placed in the 9th bit of the memory. The partially completed memory model is below (add the memory declaration)

```
module my_mem(
                    input clk,
                    input write,
                    input read,
                    input [7:0] data_in,
                    input [15:0] address,
                    output reg [7:0] data_out
);

  // Declare a 9-bit associative array using the logic data type & the key of int datatype
  <Put your declaration here>

  always @(posedge clk) begin
    if (write)
      mem_array[address] = {~^data_in, data_in};
    else if (read)
      data_out =  mem_array[address];
  end

endmodule
```

Question 3

I. Verification plan

Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
RAM_1	During a write operation, the correct data must be written to the RAM at the specified address.	Randomize address and data_in during the write operation to test a variety of addresses and data values.	Ensure all possible address ranges and data values are covered, including boundary conditions (min and max addresses).	A checker verifies that the data is correctly written into the RAM at the given address.
RAM_2	During a read operation, the correct data must be read from the RAM at the specified address.	Randomize the address and verify the data read from the RAM matches the expected data.	Ensure coverage for all addresses that were written and randomize read operations for different addresses.	A checker verifies the data read from RAM matches the expected data stored at the address.
RAM_3	The data read from RAM should be consistent with the data written to it, and parity bit (^data_in) should be computed.	Randomize data_in for write operations and compute parity to compare against expected values during read operations.	Ensure parity bit correctness is validated for different data inputs across various addresses.	A checker compares the actual data and parity bit read from the RAM against the expected data and parity generated during write.
RAM_4	After the last write operation, ensure the last value written is correctly read from the RAM before the read operation begins.	Apply a delay after the final write to ensure timing correctness between write and read operations.	Ensure that both the final write and first read operations are covered, especially at the edges of the test.	A checker ensures the last written data is stored correctly and can be read back accurately after the last write.
RAM_5	Data should remain stable when no read or write operation is triggered (read and write both set to 0).	Generate test cases where no read or write operation is performed and ensure data remains unchanged during this time.	Ensure coverage for cases where the RAM is idle and no read or write operation is happening.	A checker verifies that data_out remains stable and no unintended read or write occurs when both read and write signals are low.

II. RTL design

```
module RAM(
input clk,
input write,
input read,
input [7: 0] data_in,
input [18: 0] address,

output reg [8: 0] data_out
);

logic [8: 0] mem_array [bit [18: 0]] ;
always @(posedge clk) begin
if (write)

mem_array[address] = {^data_in, data_in};
else if (read)
data_out = mem_array [address];
end
endmodule
```

III. Test bench

```
module RAM_tb;
    // Input and Output Declaration
    logic clk, write, read;
    logic [7:0] data_in;
    logic [18:0] address;
    logic [8:0] data_out;

    localparam TESTS = 100;
    // Error & Correct Counters
    integer error_counter, correct_counter;

    bit [18:0] address_array[];
    bit [7:0] data_to_write_array[];
    bit [8:0] data_read_expect_assoc[bit [18:0]];
    bit [8:0] data_read_queue[$];
    //Design Module Instantiation
      RAM DUT(.*);
    //CLOCK GENERATION
    initial begin
    clk=0;
    forever
    #1 clk=~clk;
    end
    task stimulus_gen();
    for (int i = 0 ; i < TESTS ; i++) begin
    address_array[i] = $random;
    data_to_write_array[i] = $random;
    end
    endtask
    task golden_model();
    for (int i = 0 ; i < TESTS ; i++)
    data_read_expect_assoc[address_array[i]]= {^data_to_write_array[i], data_to_write_array[i]};
    endtask
    // Initial Values for Inputs & Main Block Operations
   initial begin
   address_array = new[TESTS];
   data_to_write_array = new[TESTS];
  // Initialize all Counters & Inputs to ZERO
   error_counter = 0;
```

```
    correct_counter = 0;
    read = 0;
    write = 0;
    address = 0;
    data_in = 0;
   // Delay 1 Clock Cycles
    repeat(1) @(negedge clk);
    stimulus_gen();
    golden_model();
   // 100 Write Operations
    write = 1;
   for (int i = 0 ; i < TESTS ; i++) begin
   @(negedge clk);
   address = address_array[i];
   data_in = data_to_write_array[i];
   end
   // Delay 1 Clock Cycles to Allow final Data to be Written in RAM
   repeat(1) @(negedge clk);
   // 100 Read Operations
     write = 0;
     read = 1;
     address_array. reverse();
     for (int i = 0 ; i < TESTS ; i++) begin
    @(negedge clk);
    address = address_array[i];
    check_result(address);
    data_read_queue. insert(i, data_out);
    end
    $display("Data Read From The Queue");
    while (data_read_queue. size())
    $display("At %0t ns: Data Read From Queue %0d", $time, data_read_queue. pop_front());
    read = 0;
    repeat(2) @(negedge clk);
    $display("%0t: At End of test error counter = %0d and correct counter = %0d", $time, error_counter, correct_counter);
    $stop;
    end
    task check_result(input bit [18: 0] Expected_Address);
    @(negedge clk);
    if (data_read_expect_assoc.exists(Expected_Address)) begin
    if (data_out !== data_read_expect_assoc[Expected_Address]) begin
    $display("%0t: Error: For Read Data %0h should be equal to %0h but it is not equal",$time, data_out, data_read_expect_assoc[address_array[Expected_Address]]);
    error_counter ++;
    end
    else
    correct_counter ++;
   end
   endtask

endmodule
```

IV. Print

```
# At 604 ns: Data Read From Queue 417
# At 604 ns: Data Read From Queue 353
# At 604 ns: Data Read From Queue 377
# At 604 ns: Data Read From Queue 219
# At 604 ns: Data Read From Queue 120
# At 604 ns: Data Read From Queue 476
# At 604 ns: Data Read From Queue 438
# At 604 ns: Data Read From Queue 192
# At 604 ns: Data Read From Queue 311
# At 604 ns: Data Read From Queue 282
# At 604 ns: Data Read From Queue 250
# At 604 ns: Data Read From Queue 442
# At 604 ns: Data Read From Queue 368
# At 604 ns: Data Read From Queue 317
# At 604 ns: Data Read From Queue 390
# At 604 ns: Data Read From Queue 297
# At 604 ns: Data Read From Queue 284
# At 604 ns: Data Read From Queue 46
# At 604 ns: Data Read From Queue 45
# At 604 ns: Data Read From Queue 136
# At 604 ns: Data Read From Queue 105
# At 604 ns: Data Read From Queue 260
# At 604 ns: Data Read From Queue 149
# At 604 ns: Data Read From Queue 202
# At 604 ns: Data Read From Queue 365
# At 604 ns: Data Read From Queue 77
# At 604 ns: Data Read From Queue 89
# At 604 ns: Data Read From Queue 147
# At 604 ns: Data Read From Queue 184
# At 604 ns: Data Read From Queue 312
# At 604 ns: Data Read From Queue 10
# At 604 ns: Data Read From Queue 78
# At 604 ns: Data Read From Queue 195
# At 604 ns: Data Read From Queue 154
# At 604 ns: Data Read From Queue 270
# At 604 ns: Data Read From Queue 298
# At 604 ns: Data Read From Queue 68
# At 604 ns: Data Read From Queue 479
# At 604 ns: Data Read From Queue 101
# At 604 ns: Data Read From Queue 237
# At 604 ns: Data Read From Queue 90
# At 604 ns: Data Read From Queue 459
# At 604 ns: Data Read From Queue 68
# At 604 ns: Data Read From Queue 435
# At 604 ns: Data Read From Queue 163
# At 604 ns: Data Read From Queue 294
```

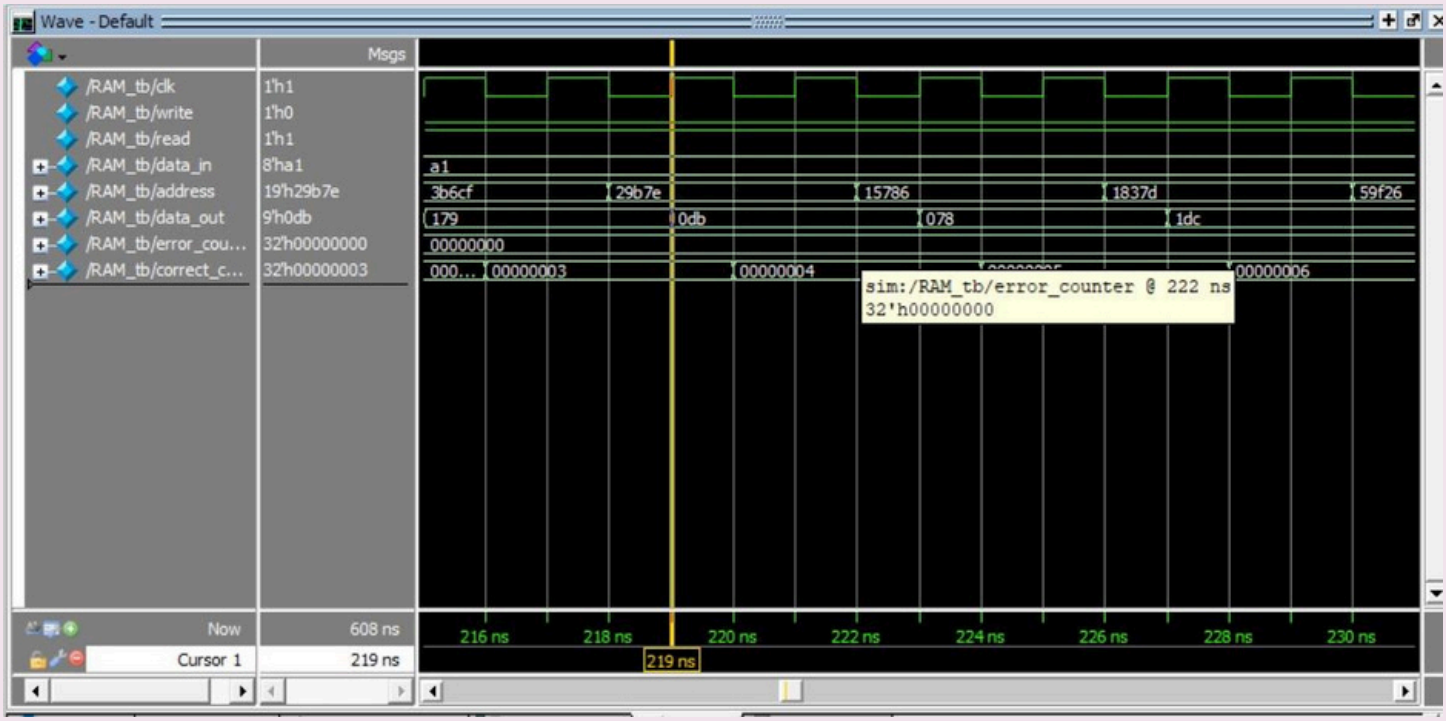
```
# At 604 ns: Data Read From Queue 156
# At 604 ns: Data Read From Queue 390
# At 604 ns: Data Read From Queue 298
# At 604 ns: Data Read From Queue 329
# At 604 ns: Data Read From Queue 120
# At 604 ns: Data Read From Queue 467
# At 604 ns: Data Read From Queue 57
# At 604 ns: Data Read From Queue 126
# At 604 ns: Data Read From Queue 317
# At 604 ns: Data Read From Queue 456
# At 604 ns: Data Read From Queue 12
# At 604 ns: Data Read From Queue 337
# At 604 ns: Data Read From Queue 464
# At 604 ns: Data Read From Queue 393
# At 604 ns: Data Read From Queue 92
# At 604 ns: Data Read From Queue 183
# At 604 ns: Data Read From Queue 399
# At 604 ns: Data Read From Queue 332
# At 604 ns: Data Read From Queue 473
# At 604 ns: Data Read From Queue 277
# At 604 ns: Data Read From Queue 58
# At 604 ns: Data Read From Queue 335
# At 604 ns: Data Read From Queue 113
# At 604 ns: Data Read From Queue 298
# At 604 ns: Data Read From Queue 430
# At 604 ns: Data Read From Queue 438
# At 604 ns: Data Read From Queue 393
# At 604 ns: Data Read From Queue 216
# At 604 ns: Data Read From Queue 394
# At 604 ns: Data Read From Queue 60
# At 604 ns: Data Read From Queue 10
# At 604 ns: Data Read From Queue 207
# At 604 ns: Data Read From Queue 430
# At 604 ns: Data Read From Queue 469
# At 604 ns: Data Read From Queue 83
# At 604 ns: Data Read From Queue 275
# At 604 ns: Data Read From Queue 413
# At 604 ns: Data Read From Queue 288
# At 604 ns: Data Read From Queue 10
# At 604 ns: Data Read From Queue 101
# At 604 ns: Data Read From Queue 189
# At 604 ns: Data Read From Queue 197
# At 604 ns: Data Read From Queue 462
# At 604 ns: Data Read From Queue 399
# At 604 ns: Data Read From Queue 119
# At 604 ns: Data Read From Queue 170
```

```
# At 604 ns: Data Read From Queue 198
# At 604 ns: Data Read From Queue 396
# At 604 ns: Data Read From Queue 317
# At 604 ns: Data Read From Queue 269
# At 604 ns: Data Read From Queue 18
# At 604 ns: Data Read From Queue 141
# At 604 ns: Data Read From Queue 99
# At 604 ns: Data Read From Queue 129
# 608: At End of test error counter = 0 and correct counter = 100
```

V. Do file

```
vlib work
vlog RAM.sv RAM_tb.sv  +cover -covercells
vsim -voptargs=+acc work.RAM_tb -cover
add wave *
coverage save RAM_tb.ucdb -onexit -du work.RAM
run -all
```

VI. Waveform



VII. Coverage Report

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

================================Branch Details================================
Branch Coverage for instance /\RAM_tb#DUT

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File RAM.sv
------------------------------------IF Branch------------------------------------
    13                                       304     Count coming in to IF
    13              1                        101
    16              1                        200
                                               3     All False Count
Branch totals: 3 hits of 3 branches = 100.00%


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

================================Statement Details================================
Statement Coverage for instance /\RAM_tb#DUT  --

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File RAM.sv
    12              1                        304
    15              1                        101
    17              1                        200
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         78        78         0   100.00%

================================Toggle Details================================
Toggle Coverage for instance /\RAM_tb#DUT  --
                                      Node      1H->0L      0L->1H  "Coverage"
                                      ----------------------------------------
                                address[0-18]           1           1      100.00
                                         clk            1           1      100.00
                                 data_in[0-7]           1           1      100.00
                                data_out[8-0]           1           1      100.00
                                        read            1           1      100.00
                                       write            1           1      100.00

Total Node Count     =         39
Toggled Node Count   =         39
Untoggled Node Count =          0
Toggle Coverage      =     100.00% (78 of 78 bins)
Total Coverage By Instance (filtered view): 100.00%
```