يوسف أحمد محمد ابراهيم

Assignment 3

Extra

Question 1:

1. Design :

```
module tb1();
    int j;
    int q[$];

    initial begin
        j = 1;
        q = '{0, 2, 5};
        q.insert(1, j);
        $display("q = %p", q);
        q.delete(1);
        $display("q = %p", q);
        q.push_front(7);
        $display("q = %p", q);
        q.push_back(9);
        $display("q = %p", q);
        j = q.pop_front();
        $display("q = %p, j = %0d", q, j);
        j = q.pop_back();
        $display("q = %p, j = %0d", q, j);
        q.reverse();
        $display("reverse q = %p", q);
        q.rsort();
        $display("rsort q = %p", q);
        q.shuffle();
        $display("shuffle q = %p" , q);

    end
endmodule
```

2. Result:

```
# q = '{0, 1, 2, 5}
# q = '{0, 2, 5}
# q = '{7, 0, 2, 5}
# q = '{7, 0, 2, 5, 9}
# q = '{0, 2, 5, 9}, j = 7
# q = '{0, 2, 5}, j = 9
# reverse q = '{5, 2, 0}
# rsort q = '{5, 2, 0}
# shuffle q = '{2, 0, 5}
```

Question 2:

1. RTL Design :

```
module adder (
    input  clk,
    input  reset,
    input  signed [3:0] A, // Input data A in 2's complement
    input  signed [3:0] B, // Input data B in 2's complement
    output reg signed [4:0] C // Adder output in 2's complement
        );

   // Register output C
   always @(posedge clk or posedge reset) begin
      if (reset)
        C <= 5'b0;
      else
        C <= A + B;
   end

endmodule
```

2. Testbench:

```
import adder_pkg::*;

module adder_tb();
    bit signed [3:0]   A;
    bit signed [3:0]   B;
    bit signed [4:0]   C;
    bit        clk;
    logic      rst;

    integer error_count, correct_count;

    localparam MAXPOS = 4'b0111;
    localparam MAXNEG = 4'b1000;

    adder dut(clk, rst, A, B, C);

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

    rand_input ri;

    initial begin
        ri = new(clk);
        A = 0;
        B = 0;
        error_count = 0;
        correct_count = 0;
        assert_reset();

        for (int i = 0; i < 200; i++) begin
            assert(ri.randomize());
            A = ri.A;
            B = ri.B;
            rst = ri.rst;
            check_result(A + B);
        end

        $display("errors: %d, success: %d", error_count, correct_count);
        $stop;
    end
```

2. Testbench:

```
45      task assert_reset();
46          rst = 1;
47          check_result(0);
48          rst = 0;
49      endtask
50
51      task check_result(logic signed [4:0] C_exp);
52          if(rst) C_exp = 0;
53          @(negedge clk);
54              if(C != C_exp) begin
55                  $display ("*** ERROR, A = %d, B = %d, C = %d", A, B, C, );
56                  error_count = error_count + 1;
57              end
58              else
59                  correct_count = correct_count + 1;
60      endtask
61
62      always @(posedge clk) begin
63          if(!rst) begin
64              ri.cg_A.sample();
65              ri.cg_B.sample();
66          end
67      end
68  endmodule
69
```

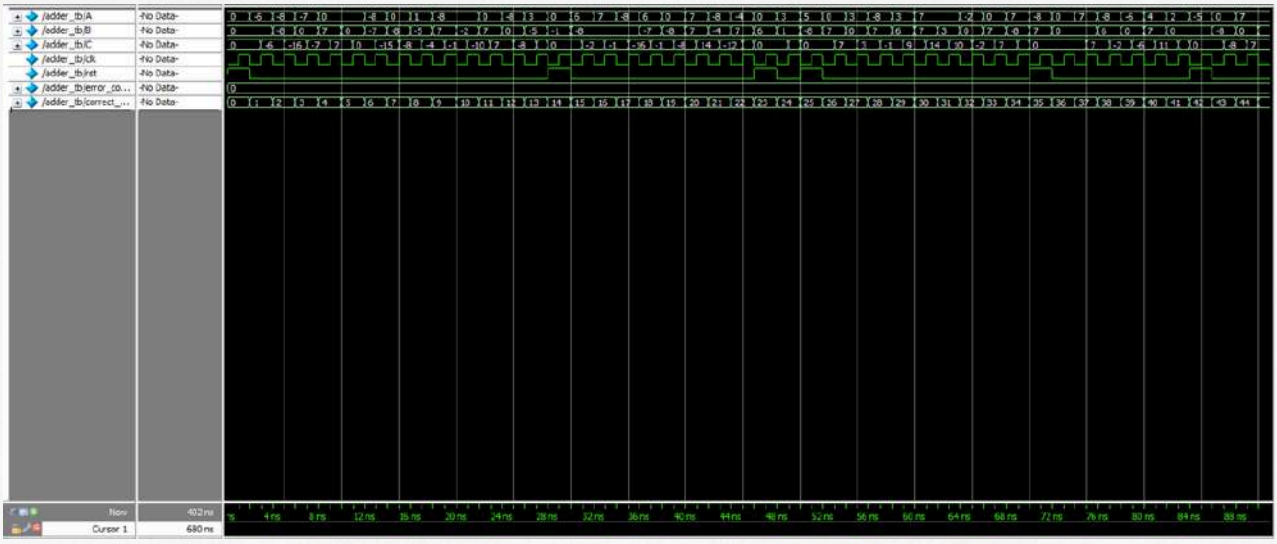
3. Verification Plan:

	Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
2	RESET_1	When the reset is asserted, output **C** should be low regardless value of **A, B.**	Directed at the start and the end of the simulation	-	**assert_reset()** task checks that output is 0 during reset.
3	RANDOMIZED	C always equal to a + b if no reset	using class to random values	**two covergroups with 2 coverpoints each, first coverpoint contain bins for ZERO MAXPOS and MAXNEG snd a default, second coverpoint contains bins for (0 > MAXPOS) (MAXPOS > MAXNEG) and (MAXNEG > MAXPOS)**	check_result task checks for the correct output.

4. Do file:

```
vlib work
vlog 2_adder_pkg.sv 2_adder.v 2_adder_tb.sv  +cover -covercells
vsim -voptargs=+acc work.adder_tb -cover
add wave *
coverage save adder_tb.ucdb -onexit
run -all
```

5. Questasim snippet:



```
# errors:            0, success:          201
# ** Note: $stop    : 2_adder_tb.sv(42)
#    Time: 402 ns  Iteration: 1  Instance: /adder_tb
# Break in Module adder_tb at 2_adder_tb.sv line 42
```

6. statement coverage:

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

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

Statement Coverage for instance /adder_tb/dut --

    Line         Item                     Count     Source
    ----         ----                     -----     ------
  File 2_adder.v
    1                                               module adder (

    2                                                   input  clk,

    3                                                   input  reset,

    4                                                   input  signed [3:0] A, // Input data A in 2's complement

    5                                                   input  signed [3:0] B, // Input data B in 2's complement

    6                                                   output reg signed [4:0] C // Adder output in 2's complement

    7                                                        );

    8

    9                                                  // Register output C

    10             1                      220          always @(posedge clk or posedge reset) begin

    11                                                    if (reset)

    12             1                       41             C <= 5'b0;

    13                                                    else

    14             1                      179             C <= A + B;

```

7. Branch coverage:

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

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

Branch Coverage for instance /adder_tb/dut

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File 2_adder.v
------------------------------------IF Branch------------------------------------
    11                                       220     Count coming in to IF
    11             1                          41           if (reset)

    13             1                         179           else

Branch totals: 2 hits of 2 branches = 100.00%
```

8. Toggle coverage:

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

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

Toggle Coverage for instance /adder_tb/dut --

                                    Node      1H->0L      0L->1H  "Coverage"
                                    ----------------------------------------
                                  A[0-3]           1           1      100.00
                                  B[0-3]           1           1      100.00
                                  C[4-0]           1           1      100.00
                                     clk           1           1      100.00
                                   reset           1           1      100.00

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

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

9. Functional coverage :

```
Covergroup Coverage:
    Covergroups                    2        na        na   100.00%
        Coverpoints/Crosses        4        na        na        na
            Covergroup Bins       12        12         0   100.00%

----------------------------------------------------------------------------------------------------
Covergroup                                          Metric       Goal       Bins    Status

----------------------------------------------------------------------------------------------------
TYPE /adder_pkg/rand_input/cg_A                     100.00%        100          -    Covered
    covered/total bins:                                   6          6          -
    missing/total bins:                                   0          6          -
    % Hit:                                          100.00%        100          -
    Coverpoint a1                                   100.00%        100          -    Covered
        covered/total bins:                               3          3          -
        missing/total bins:                               0          3          -
        % Hit:                                      100.00%        100          -
    Coverpoint a2                                   100.00%        100          -    Covered
        covered/total bins:                               3          3          -
        missing/total bins:                               0          3          -
        % Hit:                                      100.00%        100          -
Covergroup instance \/adder_pkg::rand_input::cg_A
                                                    100.00%        100          -    Covered
    covered/total bins:                                   6          6          -
    missing/total bins:                                   0          6          -
    % Hit:                                          100.00%        100          -
    Coverpoint a1                                   100.00%        100          -    Covered
        covered/total bins:                               3          3          -
        missing/total bins:                               0          3          -
        % Hit:                                      100.00%        100          -
        bin zero                                         40          1          -    Covered
        bin maxpos                                       38          1          -    Covered
        bin maxneg                                       46          1          -    Covered
        default bin range                                56                     -    Occurred
    Coverpoint a2                                   100.00%        100          -    Covered
        covered/total bins:                               3          3          -
        missing/total bins:                               0          3          -
        % Hit:                                      100.00%        100          -
        bin data0_max                                    11          1          -    Covered
        bin data_max_min                                 14          1          -    Covered
        bin data_min_max                                 10          1          -    Covered
TYPE /adder_pkg/rand_input/cg_B                     100.00%        100          -    Covered
    covered/total bins:                                   6          6          -
    missing/total bins:                                   0          6          -
    % Hit:                                          100.00%        100          -
    Coverpoint b1                                   100.00%        100          -    Covered
        covered/total bins:                               3          3          -
        missing/total bins:                               0          3          -
        % Hit:                                      100.00%        100          -
    Coverpoint b2                                   100.00%        100          -    Covered
        covered/total bins:                               3          3          -
        missing/total bins:                               0          3          -
        % Hit:                                      100.00%        100          -
Covergroup instance \/adder_pkg::rand_input::cg_B
                                                    100.00%        100          -    Covered
    covered/total bins:                                   6          6          -
    missing/total bins:                                   0          6          -
    % Hit:                                          100.00%        100          -
    Coverpoint b1                                   100.00%        100          -    Covered
        covered/total bins:                               3          3          -
        missing/total bins:                               0          3          -
        % Hit:                                      100.00%        100          -
        bin zero                                         50          1          -    Covered
        bin maxpos                                       25          1          -    Covered
        bin maxneg                                       32          1          -    Covered
        default bin range                                73                     -    Occurred
    Coverpoint b2                                   100.00%        100          -    Covered
        covered/total bins:                               3          3          -
        missing/total bins:                               0          3          -
        % Hit:                                      100.00%        100          -
        bin data0_max                                     7          1          -    Covered
        bin data_max_min                                  2          1          -    Covered
        bin data_min_max                                  5          1          -    Covered
```

Question 3:

1. RTL Design :

```
module FSM_010(clk, rst, x, y, users_count);
    parameter IDLE  = 2'b00;
    parameter ZERO  = 2'b01;
    parameter ONE   = 2'b10;
    parameter STORE = 2'b11;

    input clk, rst, x;
    output y;
    output reg [9:0] users_count;

    reg [1:0] cs, ns;

    always @(*) begin
        case (cs)
            IDLE:
                if (x)
                    ns = IDLE;
                else
                    ns = ZERO;
            ZERO:
                if (x)
                    ns = ONE;
                else
                    ns = ZERO;
            ONE:
                if (x)
                    ns = IDLE;
                else
                    ns = STORE;
            STORE:
                if (x)
                    ns = IDLE;
                else
                    ns = ZERO;
            default:    ns = IDLE;
        endcase
    end

```

1. RTL Design :

```
46      always @(posedge clk or posedge rst) begin
47          if(rst) begin
48              cs <= IDLE;
49          end
50          else begin
51              cs <= ns;
52          end
53      end
54
55      always @(posedge clk or posedge rst) begin
56          if(rst) begin
57              users_count <= 0;
58          end
59          else begin
60              if (cs == STORE)
61                  users_count <= users_count + 1;
62          end
63      end
64
65      assign y = (cs == STORE)? 1:0;
66
67  endmodule
```

2. Testbench code:

```
import FSM_pkg::*;

module FSM_010_tb();
    logic clk;
    logic rst;
    logic x;
    logic y;
    logic [9:0]users_count;

    logic y_exp;
    logic [9:0]users_count_exp;

    integer correct_count, error_count;

    fsm_transaction my_input;

    FSM_010 dut(.*);
    FSM_010_golden golden(clk, rst, x, y_exp, users_count_exp);

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


    initial begin
        correct_count = 0;
        error_count = 0;
        x = 0;

        //reset_test
        rst   = 1;
        check_result();

        //randomize test
        my_input = new(clk);
        repeat(10000) begin
            assert(my_input.randomize());
            rst = my_input.rst;
            x = my_input.x;
            check_result();
        end

        $display("*** ERROR count: %0d, CORRECT count: %0d", error_count, correct_count);
        $stop;
    end

    task check_result();
        @(negedge clk);
        if(y != y_exp || users_count != users_count_exp) begin
            $display("*** ERROR! at time %0t, y = %0d, Expected = %0d***",
                $time, y, y_exp);
            error_count++;
        end
        else
            correct_count++;

    endtask

endmodule
```

3. Package:

```
2  package FSM_pkg;
3      class fsm_transaction;
4          rand logic x;
5          rand logic rst;
6
7          constraint c1 {
8              rst dist {1 := 1, 0 := 90};
9              x dist   {0 := 67, 1:= (100 - 67)};
10         }
11
12         covergroup cg_x (ref logic clk) @(posedge clk);
13             coverpoint x {
14                 bins tr = (0 =>1);
15             }
16         endgroup
17
18         function new(ref logic clk);
19             cg_x = new(clk);
20         endfunction
21     endclass
22 endpackage
```

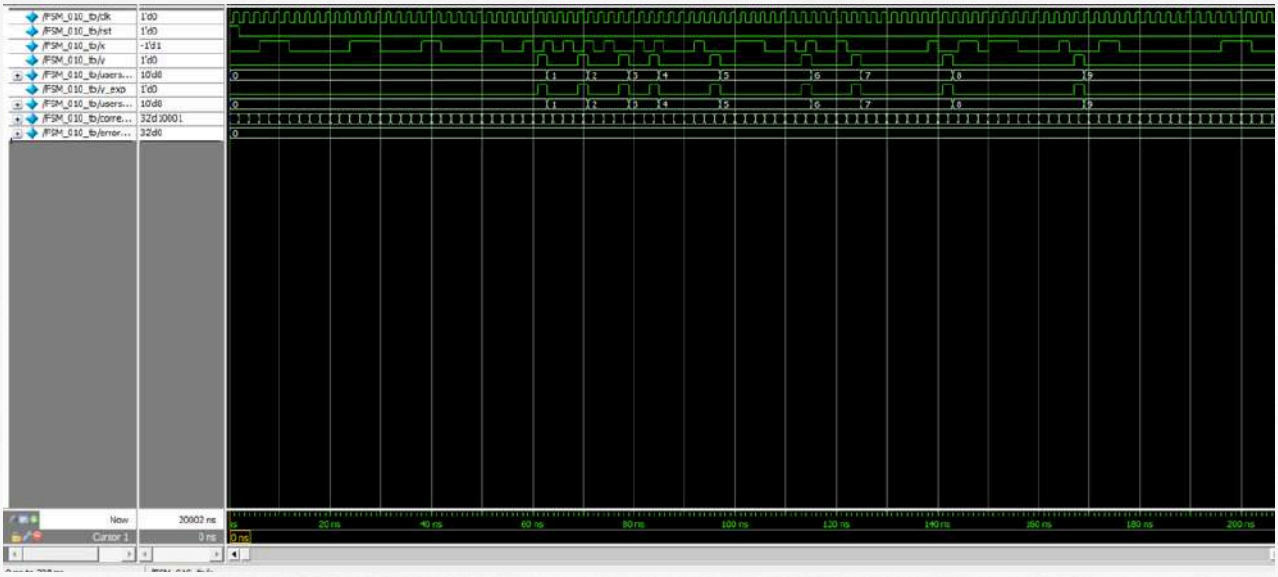
4. Verification Plan:

	Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
2	RESET test	When the reset is asserted, outputs should be low.	Directed at the start and the end of the simulation, and called in radomization step.	-	**check_result()** task checks that output is 0 during reset.
3	RANDOMIZE test	Output of the dut design is equal to the output of the golden model	Randomized test using rand_stimuls class with constraints 90% deactive reset, 67% x = 1.	**one Cover group with one coerpint on x has one bin of transition 0 => 1**	**check_result()** verifies correct output using golden model.

5. Do file:

```
vlib work
vlog 3_golden_model.sv 3_FSM_pkg.sv 3_FSM_010.v 3_FSM_tb.sv +cover -covercells
vsim -voptargs=+acc work.FSM_010_tb -cover
add wave *
coverage save 3_FSM_tb.ucdb -onexit -du work.FSM_010
coverage exclude -du FSM_010 -togglenode {users_count[6]}
coverage exclude -du FSM_010 -togglenode {users_count[7]}
coverage exclude -du FSM_010 -togglenode {users_count[8]}
coverage exclude -du FSM_010 -togglenode {users_count[9]}

run -all
```

6. Questasim wave:



```
# *** ERROR count: 0, CORRECT count: 10001
# ** Note: $stop    : 3_FSM_tb.sv(46)
#    Time: 20002 ns  Iteration: 1  Instance: /FSM_010_tb
# Break in Module FSM_010_tb at 3_FSM_tb.sv line 46
```

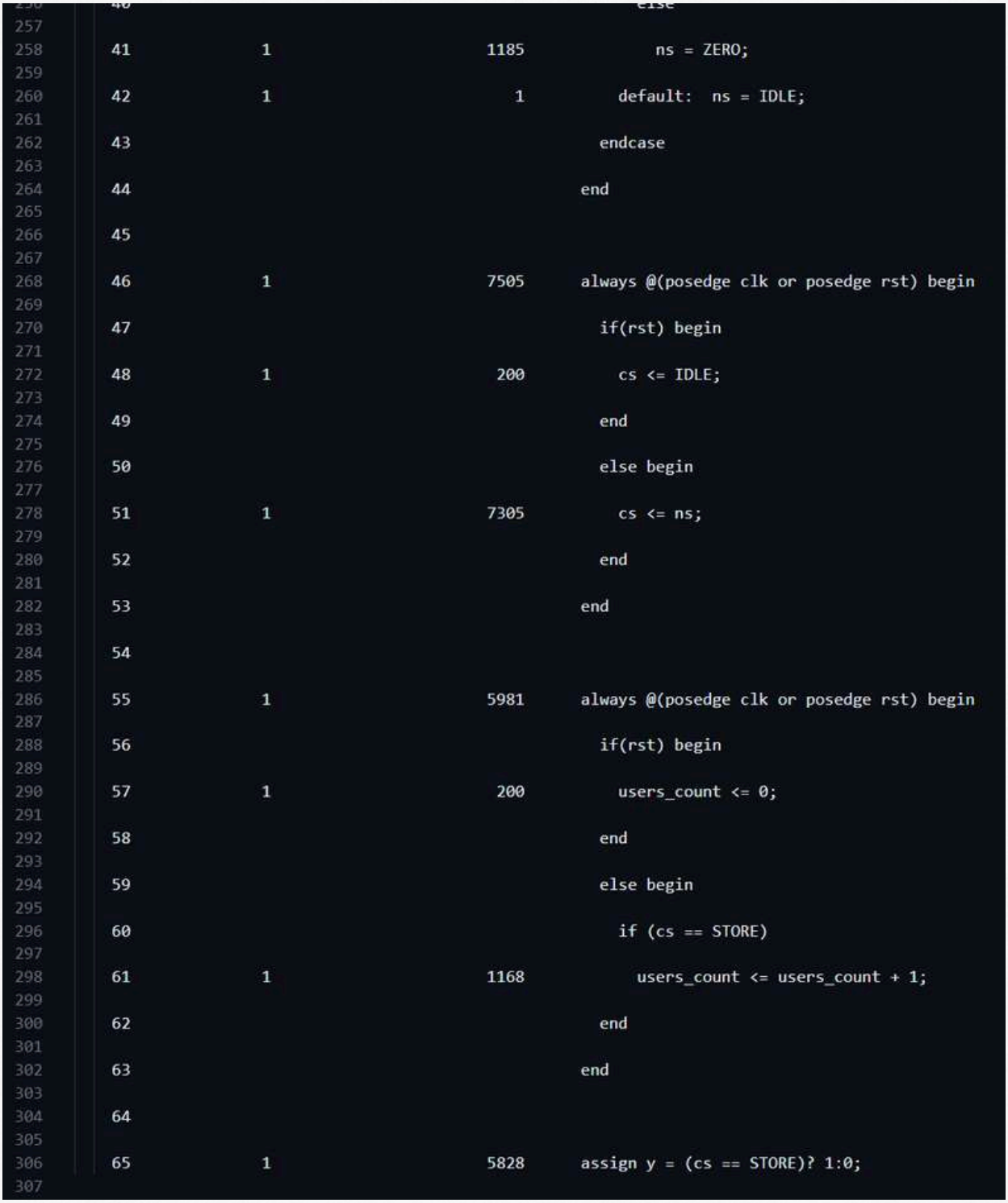
7. Statement coverage:

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

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

Statement Coverage for instance /\FSM_010_tb#dut  --

    Line         Item                     Count     Source
    ----         ----                     -----     ------
  File 3_FSM_010.v
    8                                              module FSM_010(clk, rst, x, y, users_count);

    9                                               parameter IDLE  = 2'b00;

    10                                              parameter ZERO  = 2'b01;

    11                                              parameter ONE   = 2'b10;

    12                                              parameter STORE = 2'b11;

    13

    14                                              input clk, rst, x;

    15                                              output y;

    16                                              output reg [9:0] users_count;

    17

    18                                              reg [1:0] cs, ns;

    19

    20             1                10245           always @(*) begin

    21                                               case (cs)

    22                                                 IDLE:

    23                                                   if (x)

    24             1                 1014                ns = IDLE;

    25                                                 else

    26             1                 1073                ns = ZERO;

    27                                                 ZERO:

    28                                                   if (x)

    29             1                 1785                ns = ONE;

    30                                                 else

    31             1                 1821                ns = ZERO;

    32                                                 ONE:

    33                                                   if (x)

    34             1                 1767                ns = IDLE;

    35                                                 else

    36             1                 1195                ns = STORE;

    37                                                 STORE:

    38                                                   if (x)

    39             1                  404                ns = IDLE;

    40                                                 else
```

7. Statement coverage:

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

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

Statement Coverage for instance /\FSM_010_tb#dut  --

    Line         Item                     Count     Source
    ----         ----                     -----     ------
  File 3_FSM_010.v
    8                                              module FSM_010(clk, rst, x, y, users_count);

    9                                               parameter IDLE  = 2'b00;

    10                                              parameter ZERO  = 2'b01;

    11                                              parameter ONE   = 2'b10;

    12                                              parameter STORE = 2'b11;

    13

    14                                              input clk, rst, x;

    15                                              output y;

    16                                              output reg [9:0] users_count;

    17

    18                                              reg [1:0] cs, ns;

    19

    20             1                10245           always @(*) begin

    21                                               case (cs)

    22                                                 IDLE:

    23                                                   if (x)

    24             1                 1014                ns = IDLE;

    25                                                 else

    26             1                 1073                ns = ZERO;

    27                                                 ZERO:

    28                                                   if (x)

    29             1                 1785                ns = ONE;

    30                                                 else

    31             1                 1821                ns = ZERO;

    32                                                 ONE:

    33                                                   if (x)

    34             1                 1767                ns = IDLE;

    35                                                 else

    36             1                 1195                ns = STORE;

    37                                                 STORE:

    38                                                   if (x)

    39             1                  404                ns = IDLE;

    40                                                 else
```

7. Statement coverage:

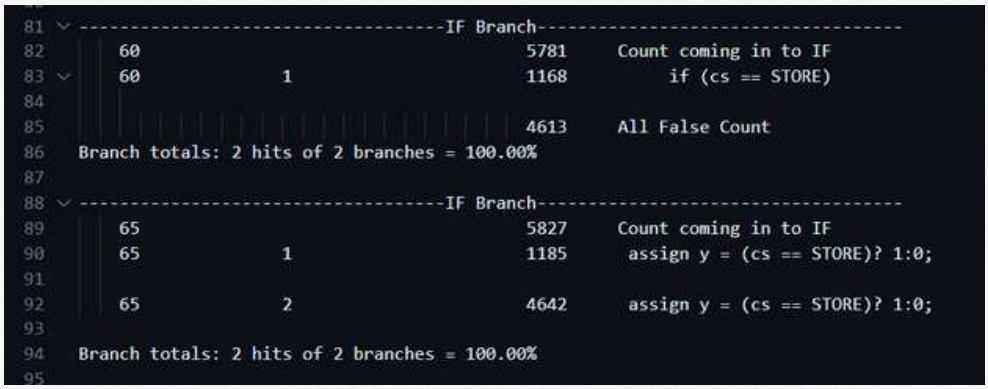


41	1	1185	ns = ZERO;
42	1	1	default: ns = IDLE;
43			endcase
44			end
45			
46	1	7505	always @(posedge clk or posedge rst) begin
47			if(rst) begin
48	1	200	cs <= IDLE;
49			end
50			else begin
51	1	7305	cs <= ns;
52			end
53			end
54			
55	1	5981	always @(posedge clk or posedge rst) begin
56			if(rst) begin
57	1	200	users_count <= 0;
58			end
59			else begin
60			if (cs == STORE)
61	1	1168	users_count <= users_count + 1;
62			end
63			end
64			
65	1	5828	assign y = (cs == STORE)? 1:0;

8. Branch coverage:

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

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

Branch Coverage for instance /\FSM_010_tb#dut

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File 3_FSM_010.v
------------------------------------CASE Branch------------------------------------
    21                                     10245     Count coming in to CASE
    22             1                        2087          IDLE:

    27             1                        3606          ZERO:

    32             1                        2962          ONE:

    37             1                        1589          STORE:

    42             1                           1          default:  ns = IDLE;

Branch totals: 5 hits of 5 branches = 100.00%

-----------------------------------IF Branch-----------------------------------
    23                                      2087     Count coming in to IF
    23             1                        1014           if (x)

    25             1                        1073           else

Branch totals: 2 hits of 2 branches = 100.00%

-----------------------------------IF Branch-----------------------------------
    28                                      3606     Count coming in to IF
    28             1                        1785           if (x)

    30             1                        1821           else

Branch totals: 2 hits of 2 branches = 100.00%

-----------------------------------IF Branch-----------------------------------
    33                                      2962     Count coming in to IF
    33             1                        1767           if (x)

    35             1                        1195           else

Branch totals: 2 hits of 2 branches = 100.00%

-----------------------------------IF Branch-----------------------------------
    38                                      1589     Count coming in to IF
    38             1                         404           if (x)

    40             1                        1185           else

Branch totals: 2 hits of 2 branches = 100.00%

-----------------------------------IF Branch-----------------------------------
    47                                      7505     Count coming in to IF
    47             1                         200        if(rst) begin

    50             1                        7305        else begin

Branch totals: 2 hits of 2 branches = 100.00%

-----------------------------------IF Branch-----------------------------------
    56                                      5981     Count coming in to IF
    56             1                         200        if(rst) begin

    59             1                        5781        else begin

Branch totals: 2 hits of 2 branches = 100.00%
```

8. Branch coverage:



9. FSM covergae:

```
139
140  ================================FSM Details================================
141
142  FSM Coverage for instance /\FSM_010_tb#dut  --
143
144  FSM_ID: cs
145      Current State Object : cs
146      ---------------------
147      State Value MapInfo :
148      ---------------------
149  Line          State Name               Value
150  ----          ----------               -----
151    22                IDLE                   0
152    27                ZERO                   1
153    32                 ONE                   2
154    37               STORE                   3
155      Covered States :
156      ----------------
157                     State          Hit_count
158                     -----          ---------
159                      IDLE               1517
160                      ZERO               3036
161                       ONE               1767
162                     STORE               1185
163      Covered Transitions :
164      --------------------
165  Line           Trans_ID          Hit_count          Transition
166  ----           --------          ---------          ----------
167    26                  0               1053          IDLE -> ZERO
168    29                  1               1767          ZERO -> ONE
169    48                  2                 54          ZERO -> IDLE
170    36                  3               1185          ONE -> STORE
171    34                  4                582          ONE -> IDLE
172    41                  5                768          STORE -> ZERO
173    39                  6                417          STORE -> IDLE
174
175
176      Summary                      Bins      Hits    Misses  Coverage
177      -------                      ----      ----    ------  --------
178          FSM States                 4         4         0   100.00%
179          FSM Transitions            7         7         0   100.00%
```

10. Toggle coverage:

```
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         28        28         0   100.00%

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

Toggle Coverage for instance /\FSM_010_tb#dut  --

                                          Node      1H->0L      0L->1H  "Coverage"
                                          ----------------------------------------
                                           clk           1           1      100.00
                                        cs[1-0]          1           1      100.00
                                        ns[1-0]          1           1      100.00
                                           rst           1           1      100.00
                                 users_count[5-0]        1           1      100.00
                                             x           1           1      100.00
                                             y           1           1      100.00

Total Node Count     =         14
Toggled Node Count   =         14
Untoggled Node Count =          0

Toggle Coverage      =     100.00% (28 of 28 bins)


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

10. Functional coverage:

```
920  COVERGROUP COVERAGE:
921  ----------------------------------------------------------------------------------------------------
922  Covergroup                                          Metric       Goal       Bins    Status
923
924  ----------------------------------------------------------------------------------------------------
925   TYPE /FSM_pkg/fsm_transaction/cg_x                100.00%        100          -    Covered
926      covered/total bins:                                  1          1          -
927      missing/total bins:                                  0          1          -
928      % Hit:                                          100.00%        100          -
929      Coverpoint x                                    100.00%        100          -    Covered
930          covered/total bins:                              1          1          -
931          missing/total bins:                              0          1          -
932          % Hit:                                      100.00%        100          -
933   Covergroup instance \/FSM_pkg::fsm_transaction::cg_x
934                                                       100.00%        100          -    Covered
935      covered/total bins:                                  1          1          -
936      missing/total bins:                                  0          1          -
937      % Hit:                                          100.00%        100          -
938      Coverpoint x                                    100.00%        100          -    Covered
939          covered/total bins:                              1          1          -
940          missing/total bins:                              0          1          -
941          % Hit:                                      100.00%        100          -
942          bin tr                                        2209          1          -    Covered
943
944  TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1
```

Question 4:

1. Design:

```
module mem();
    logic [23:0] mem [bit[19:0]];

    initial begin
        mem[20'h00000] = 24'hA50400; // reset
        mem[20'h00400] = 24'h123456; // instruction1
        mem[20'h00401] = 24'h789ABC; // instruction2
        mem[20'hFFFFF] = 24'h0F1E2D; //isr

        $display("Number of elements in memory = %0d", mem.num());
        foreach(mem[i]) begin
            $display("mem[%0h] = %0h", i, mem[i]);
        end
    end
endmodule
```

2. Result:

```
# Number of elements in memory = 4
# mem[0] = a50400
# mem[400] = 123456
# mem[401] = 789abc
# mem[fffff] = f1e2d
```

Question 5:

1. Design:

```
module struct_e();

    typedef bit [6:0] ubyte_t;

    typedef struct {
        ubyte_t header;
        ubyte_t cmd;
        ubyte_t data;
        ubyte_t crc;
    } packet;

    packet my_packet;

    initial begin
        my_packet.header = 7'h5A;
        $display("my_packet.header = %h", my_packet.header);
        $display("my_packet.header = %p", my_packet);
    end
endmodule
```

2.Result:

```
VSIM 23> run -all
# my_packet.header = 5a
# my_packet.header = '{header:90, cmd:0, data:0, crc:0}
```