يوسف أحمد محمد ابراهيم

Assignment 1

Extra

Question 1:

1. Design :

```
1   module dff(clk, rst, d, q, en);
2   parameter USE_EN = 1;
3   input clk, rst, d, en;
4   output reg q;
5
6   always @(posedge clk) begin
7       if (rst)
8           q <= 0;
9       else
10          if(USE_EN) begin
11              if (en)
12                  q <= d;
13          end
14          else
15              q <= d;
16  end
17
18  endmodule
```

2. verification plan :

	Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
2	RESET_TEST	When the reset is asserted, outputs **C** should be low.	**assert_reset()** task that asserts rst to1, wait then deassert rst to 0 at the begining and end of the simulation.	-	**check_result()** task checks that output is 0 during reset.
3	WITH_EN_TEST	Output depends on **en** value, if en = 1, **q = D.**	Directed test at en = 0, and at en = 1	-	**check_result()** checks the value of **q.**
4	WITHOUT_EN_TEST	OUtput **q = D** whatever the value of en is.	Directed test at en = 0, and at en = 1	-	**check_result()** checks the value of q.

3. Testbench 1 (USE_EN = 1) :

```
module DFF_tb1();
    logic clk;
    logic rst;
    logic d;
    logic q;
    logic en;

    logic q_exp;

    integer correct_count, error_count;

    dff dut(clk, rst, d, q, en);

    always @(posedge clk, posedge rst) begin
        if(rst)
            q_exp <= 0;
        else if(en)
            q_exp = d;
    end

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

    initial begin
        correct_count = 0;
        error_count = 0;
        d = 0;
        en = 0;
        //reset test
        rst = 1;
        check_result();
        rst = 0;

        //WITH_EN check
        d = 1;
        check_result();
        en = 1;
        check_result();

        d = 0;
        en = 0;
        check_result();

        rst = 1;
        check_result();
        rst = 0;


        $display("*** errors: %0d, success: %0d ***", error_count, correct_count);
        $stop;
    end

    task check_result();
        @(negedge clk);
        if(q != q_exp) begin
            error_count = error_count +1;
            $display ("*** ERROR! D = %0d,en = %0d, rst = %0d, q = %0d ***", d, en, rst, q);
        end
        else
            correct_count = correct_count +1;
    endtask
endmodule
```

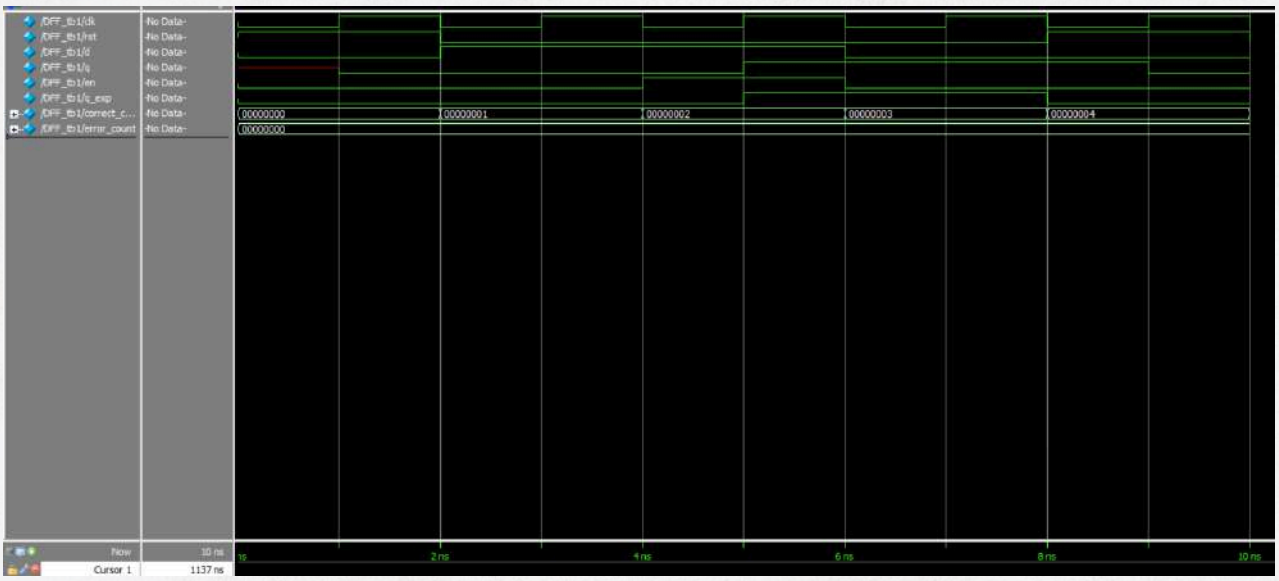
3. Testbench (USE_EN = 0) :

```
module DFF_tb2();
    logic clk;
    logic rst;
    logic d;
    logic q;
    logic en;

    logic q_exp;

    integer correct_count, error_count;

    dff #(.USE_EN(0)) dut2(clk, rst, d, q, en);

    always @(posedge clk, posedge rst) begin
        if(rst)
            q_exp <= 0;
        else
            q_exp = d;
    end

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

    initial begin
        correct_count = 0;
        error_count = 0;
        d = 0;
        en = 0;
        //reset test
        rst = 1;
        check_result();
        rst = 0;

        //WITH_EN check
        d = 1;
        check_result();
        en = 1;
        check_result();

        d = 0;
        en = 0;
        check_result();

        rst = 1;
        check_result();
        rst = 0;


        $display("*** errors: %0d, success: %0d ***", error_count, correct_count);
        $stop;
    end

    task check_result();
        @(negedge clk);
        if(q != q_exp) begin
            error_count = error_count +1;
            $display ("*** ERROR! D = %0d,en = %0d, rst = %0d, q = %0d ***", d, en, rst, q);
        end
        else
            correct_count = correct_count +1;
    endtask
endmodule
```

4. Do file (*changing vsim instance name and coverage save to the testbench to simulate*) :

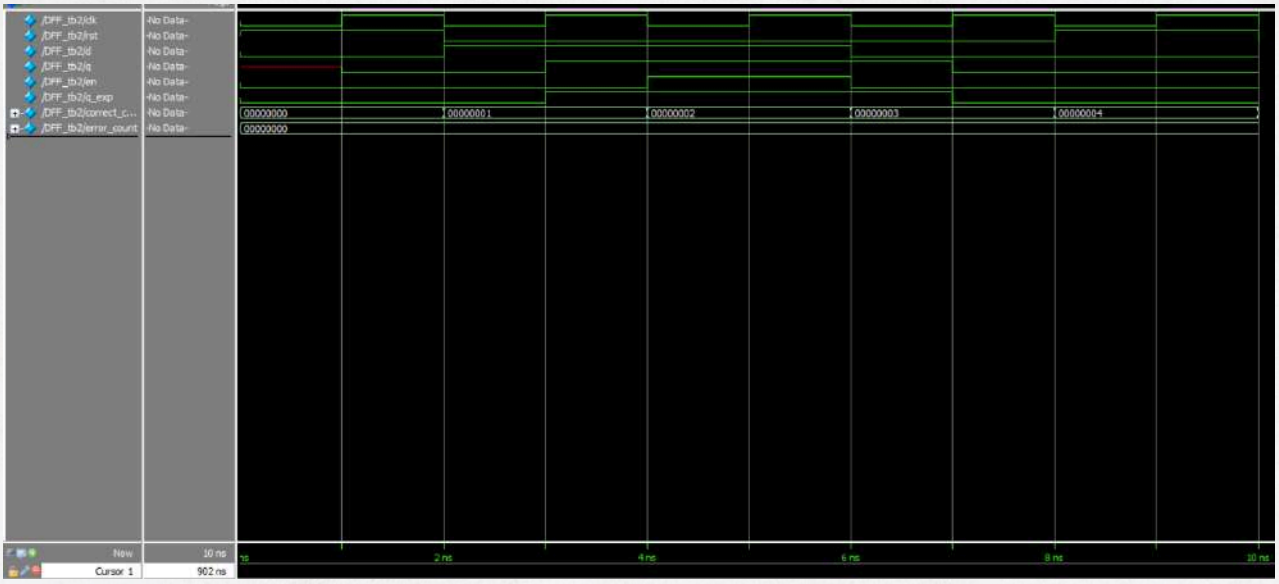
```
vlib work
vlog DFF.v DFF_tb.sv  +cover -covercells
vsim -voptargs=+acc work.DFF_tb2 -cover
add wave *
coverage save 3_ALU_tb2.ucdb -onexit -du work.dff
run -all
```

5. Qesta sim wave snippets (First TB) :




```
# *** errors: 0, success: 5 ***
# ** Note: $stop    : DFF_tb.sv(53)
#    Time: 10 ns  Iteration: 1  Instance: /DFF_tb1
# Break in Module DFF_tb1 at DFF_tb.sv line 53
```

5. Qesta sim wave snippets (Second TB) :




```
# *** errors: 0, success: 5 ***
# ** Note: $stop    : DFF_tb.sv(120)
#    Time: 10 ns  Iteration: 1  Instance: /DFF_tb2
# Break in Module DFF_tb2 at DFF_tb.sv line 120
```

6. Statement coverage report :

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

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

Statement Coverage for instance /\work.dff  --

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File DFF.v
    1                                                module dff(clk, rst, d, q, en);

    2                                                parameter USE_EN = 1;

    3                                                input clk, rst, d, en;

    4                                                output reg q;

    5

    6               1                         10     always @(posedge clk) begin

    7                                                   if (rst)

    8               1                          4          q <= 0;

    9                                                   else

    10                                                     if(USE_EN) begin

    11                                                        if (en)

    12              1                          1               q <= d;

    13                                                    end

    14                                                    else

    15              1                          3             q <= d;

    72
```

7. Branch coverage report :

```
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         4         4         0   100.00%

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

Branch Coverage for instance /\work.dff

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File DFF.v
------------------------------------IF Branch------------------------------------
    7                                        10     Count coming in to IF
    7              1                          4        if (rst)

    11             1                          1              if (en)

                                              2     All False Count
    9              1                          3        else

Branch totals: 4 hits of 4 branches = 100.00%
```

8. Toggle coverage report :

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

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

Toggle Coverage for instance /\work.dff  --

                                                    Node      1H->0L      0L->1H  "Coverage"
                                                    ----------------------------------------
                                                     clk           2           2      100.00
                                                       d           2           2      100.00
                                                      en           2           2      100.00
                                                       q           2           2      100.00
                                                     rst           2           2      100.00

Total Node Count     =          5
Toggled Node Count   =          5
Untoggled Node Count =          0

Toggle Coverage      =     100.00% (10 of 10 bins)


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