Assignment 1

Program:

Course Code: CSE412

Course Name: Digital Verification

**Omar Mohamed Omar AbdelSamea**

1700903

Ain Shams University

Faculty of Engineering

Spring Semester – 2022

MODULES

There are 2 modules multimode counter and game

1. **Multimode counter**: counter with 2 control bits that selects one of four modes

    .   Counting up by 1

    a. Counting up by 2

    b. Counting down by 1

    c. Counting down 2

2. **Game**: main module responsible for checking the counter value and sets WINNER signal high when the counter value reaches all 1's then it adds one on the WINNER count register, sets LOSER signal high when the counter value reaches all 0's then it adds one on the LOSER count register. When WINNER count or LOSER count reaches value 15 the module sets GAMEOVER signal as high immediately.
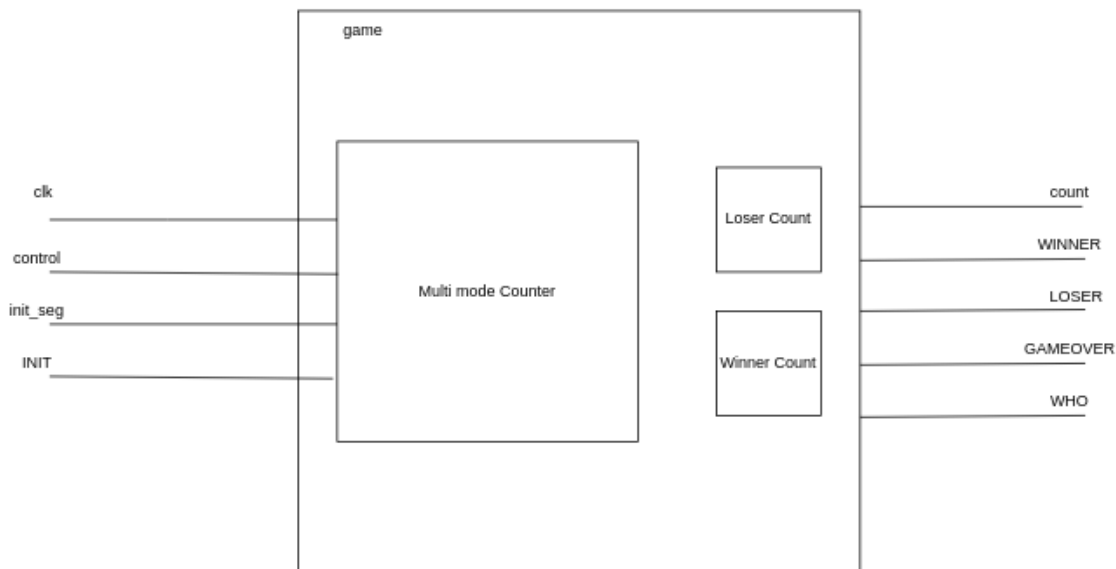
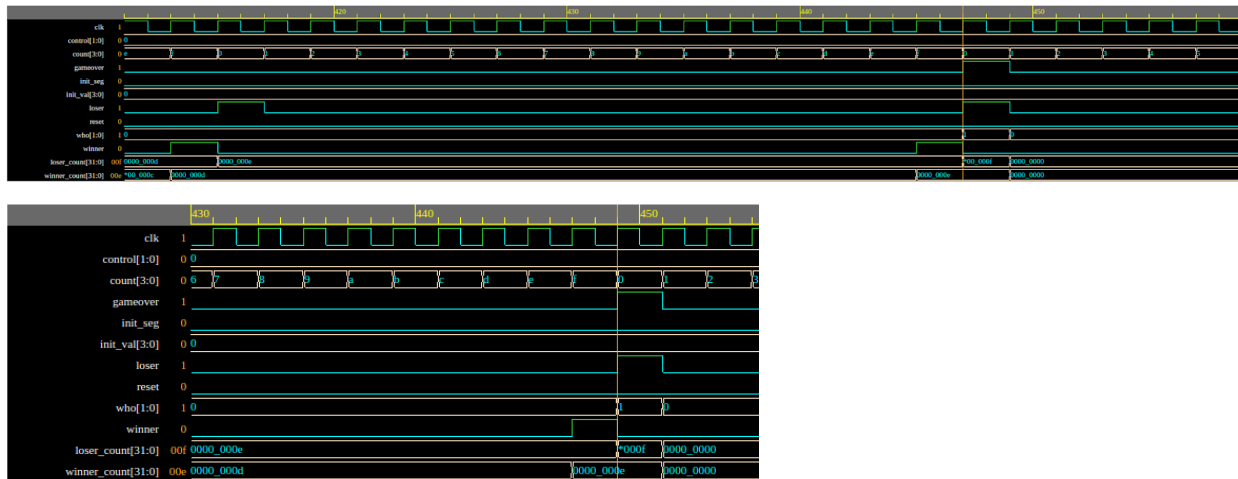### I. DOES YOUR DESIGN NEED A CLOCK?

Yes, the design needs a clock for the multimode counter.

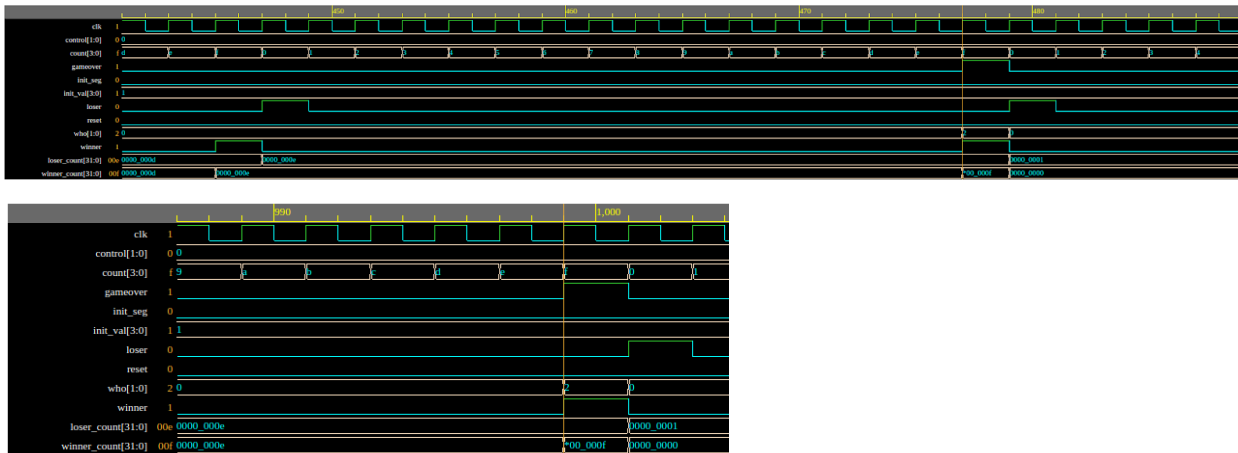### II. DOES IT NEED AN ASYNCHRONOUS RESET? A SYNCHRONOUS RESET?

Reset is asynchronous to give the ability to reset at any point of time without waiting for posedge of clock.
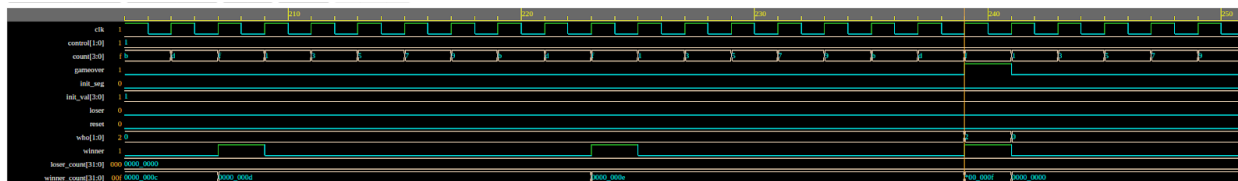
TESTBENCH

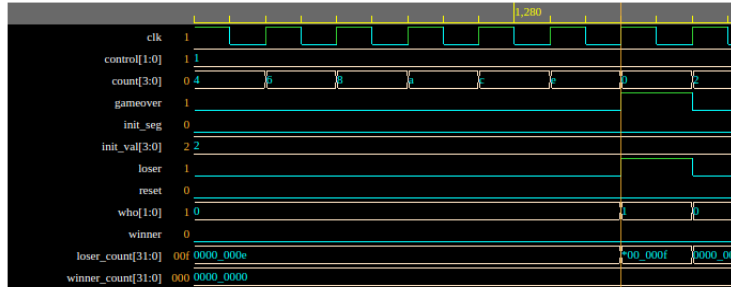**Scenario 1**: Set Initial value to 0 to give loser an advantage with ctrl 0

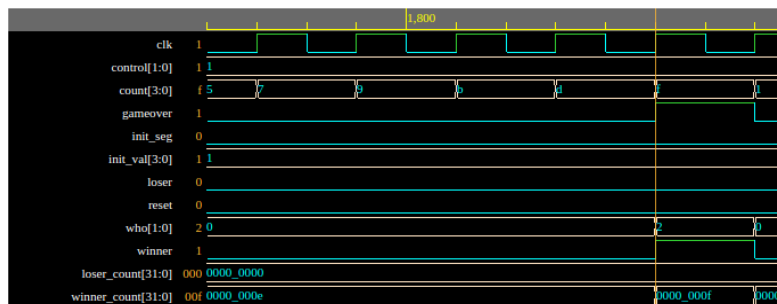**Scenario 2**: Set Initial value to 1 to give winner an advantage with ctrl 0

Scenario 3: Set Initial value to 2 to give loser an advantage with ctrl 1

Scenario 4: Set Initial value to 1 to give winner an advantage with ctrl 1

Scenario 5: Set Initial value to 1 to give loser an advantage with ctrl 2

4

Scenario 6: Set Initial value to 15 to give winner an advantage with ctrl 2

Scenario 7: Set Initial value to 0 to give loser an advantage with ctrl 3

Scenario 8: Set Initial value to 15 to give winner an advantage with ctrl 3
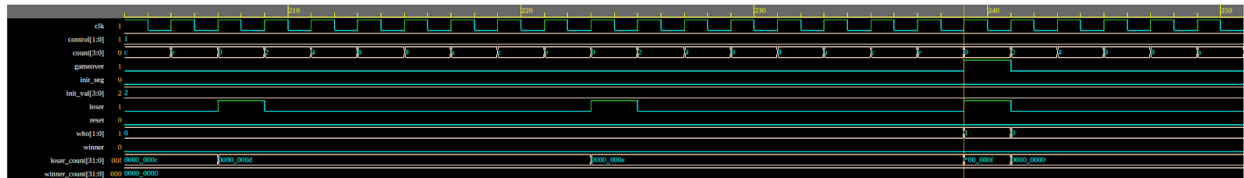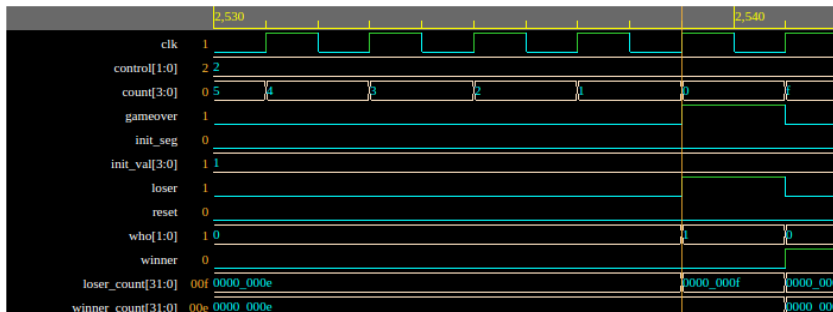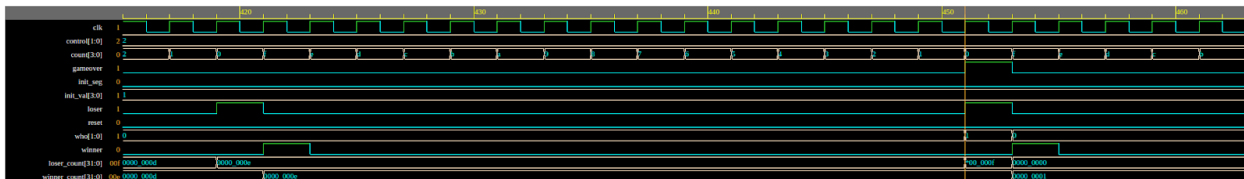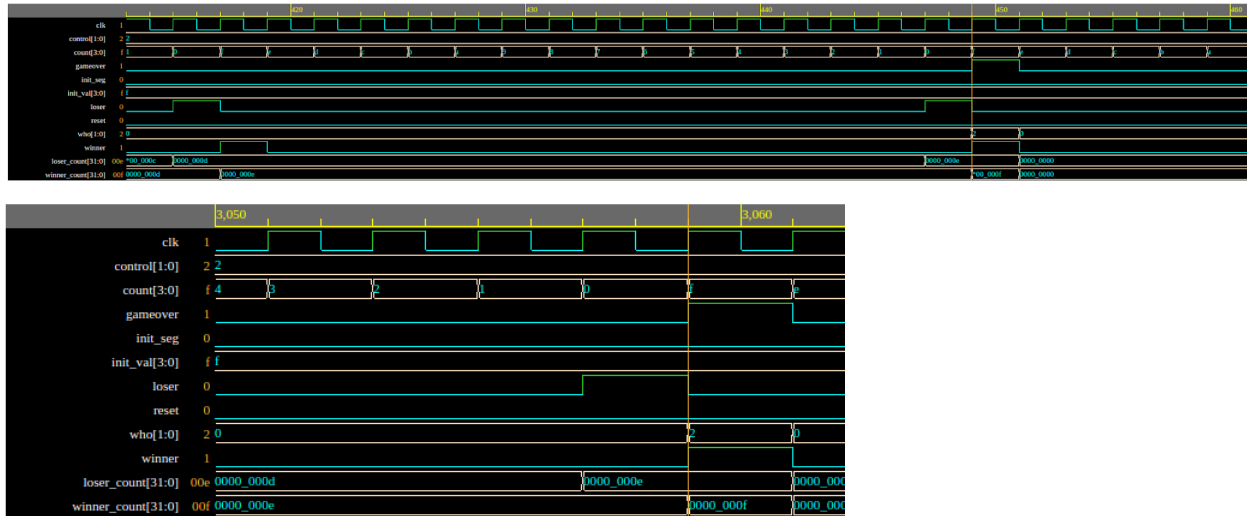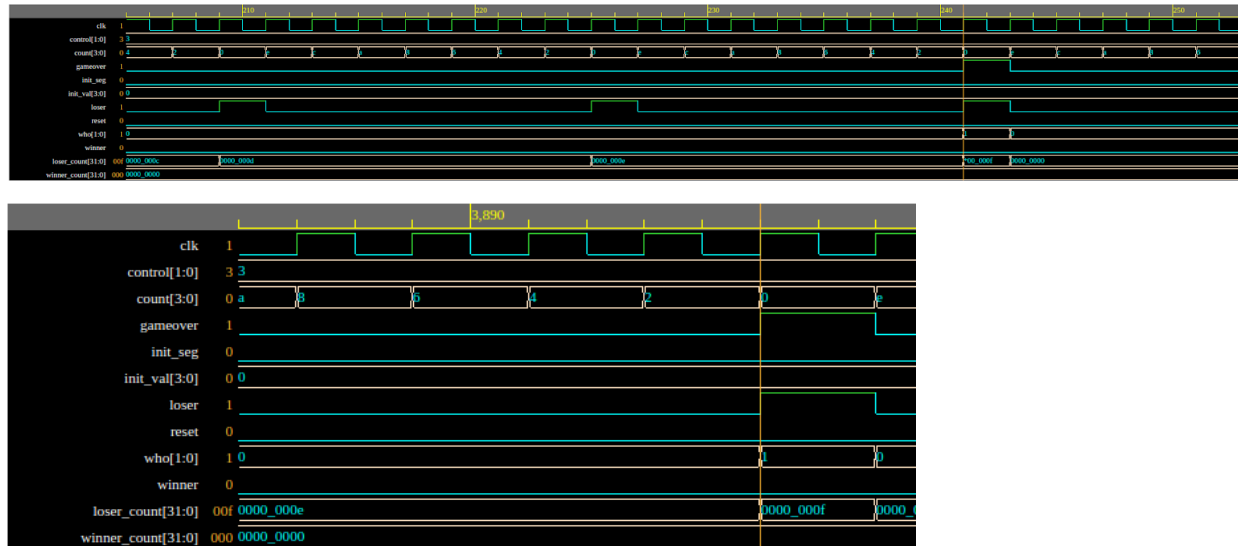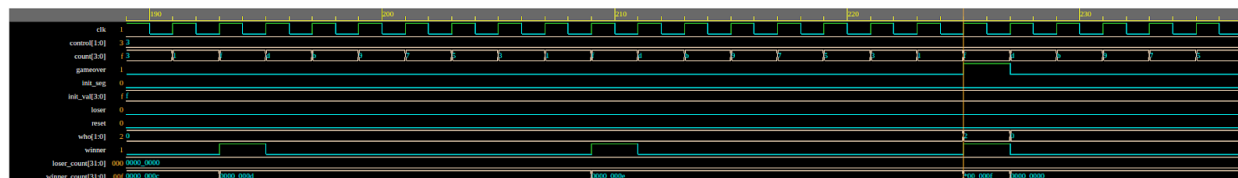
Reset example

Initial value load example

```
module game_tb#(DATA_WIDTH=4);
  parameter CLK = 1;
  bit clk;
  bit reset;
  bit [1:0] control = 2'b00;
  bit init_seg = 0;
  bit [DATA_WIDTH - 1:0] init_val;

  wire [DATA_WIDTH - 1:0] count;
  wire loser;
  wire winner;
  wire gameover;
  wire [1:0] who;

  initial begin
    clk = 1'b0;
    forever #CLK clk = ~clk;
  end

  initial begin
    /* Scenario 1: Set Initial value to 0 to give loser an advantage with
ctrl 0*/
    control = 2'b00;
    init_seg = 1; init_val = 0;
    #2 init_seg = 0;

    #500 reset = 1;
    #20 reset = 0;

    /* Scenario 2: Set Initial value to 1 to give winner an advantage with
ctrl 0*/
    init_seg = 1; init_val = 1;
    #2 init_seg = 0;

    #500 reset = 1;
    #20 reset = 0;

    control = 2'b01;
```

```
    /* Scenario 3: Set Initial value to 2 to give loser an advantage with
ctrl 1*/
    init_seg = 1; init_val = 2;
    #2 init_seg = 0;


    #500 reset = 1;
    #20 reset = 0;


    /* Scenario 4: Set Initial value to 1 to give winner an advantage with
ctrl 1*/
    init_seg = 1; init_val = 1;
    #2 init_seg = 0;


    #500 reset = 1;
    #20 reset = 0;


    control = 2'b10;
    /* Scenario 5: Set Initial value to 1 to give loser an advantage with
ctrl 2*/
    init_seg = 1; init_val = 1;
    #2 init_seg = 0;


    #500 reset = 1;
    #20 reset = 0;


    /* Scenario 6: Set Initial value to 15 to give winner an advantage
with ctrl 2*/
    init_seg = 1; init_val = 15;
    #2 init_seg = 0;


    #500 reset = 1;
    #20 reset = 0;


    control = 2'b11;
    /* Scenario 7: Set Initial value to 15 to give loser an advantage with
ctrl 3*/
    init_seg = 1; init_val = 15;
    #2 init_seg = 0;
```

```verilog
    #500 reset = 1;
    #20 reset = 0;


    /* Scenario 8: Set Initial value to 0 to give winner an advantage with
ctrl 3*/
    init_seg = 1; init_val = 0;
    #2 init_seg = 0;

end

  game dut(
    .clk (clk),
    .reset (reset),
    .control (control),
    .count (count),
    .init_seg (init_seg),
    .INIT (init_val),
    .WINNER (winner),
    .LOSER (loser),
    .GAMEOVER (gameover),
    .WHO (who)
  );

  initial begin
    $dumpfile("dump.vcd");
    $dumpvars;
    #5000 $finish;
  end
endmodule
```