**Code:**

* **Design Module**

// specifications : count (0-15)

// enable = 1----- counting start

// synchronous active high reset

//count\_load == 1 1bit

//load\_value [3:0] start counting from load\_value

// Design UP counter with load

// Task : UP-Down counter

module UpDown\_Counter(input wire clk,

input wire rst,

input wire enable,

input wire up\_oper, // up operation @ 1

input wire count\_load, // if count\_load 1 then load\_value

input wire [3:0] load\_value,

output wire [3:0] count);

reg [3:0] temp\_cnt;

always@(posedge clk , posedge rst)

begin

if(rst)

temp\_cnt <= 0;

else if(count\_load)

temp\_cnt <= load\_value;

else if(enable)

begin

if (up\_oper)

begin

if(temp\_cnt == load\_value)

temp\_cnt <= 0;

else

temp\_cnt <= temp\_cnt + 1;

end

else

begin

if(temp\_cnt == load\_value)

temp\_cnt <= 0;

else

temp\_cnt <= temp\_cnt - 1;

end

end

else

temp\_cnt <= temp\_cnt;

end

assign count = temp\_cnt;

endmodule

* **TestBench**

// Code your testbench here

// or browse Examples

module tb();

reg clk,reset,enable,up\_oper,count\_load;

reg [3:0] load\_value;

wire [3:0] count;

UpDown\_Counter dut(.clk(clk),.rst(reset),.enable(enable),.up\_oper(up\_oper),.count\_load(count\_load),.load\_value(load\_value),.count(count));

always #5 clk = ~clk;

initial

begin

clk <= 0;

repeat(5);

reset <= 0 ;

enable <= 1;

#10

up\_oper <= 1;

count\_load <= 1;

load\_value <= 3'b100;

#10

count\_load <= 0;

#10

#10

#10

#10

up\_oper <= 0;

#10

#10;

end

// only for waveform in eda playground

// initial

// begin

// $dumpfile("dump.vcd");

// $dumpvars;

// #100

// $finish;

// end

endmodule

**Output:**

