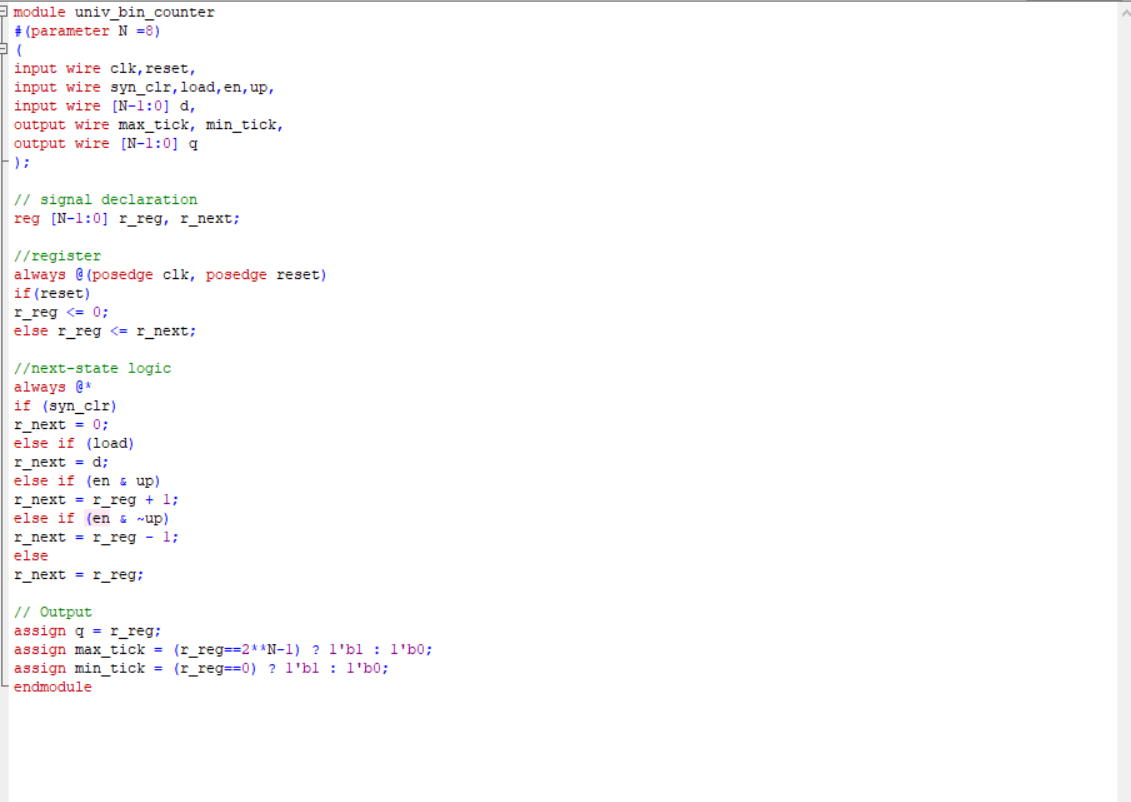
HW 11 – Universal counter

**Verilog**



**Testbench**

`timescale 1ns/10ps

module bin\_counter\_tb();

localparam T =20; //clock period

reg clk,reset;

reg syn\_clr, load, en, up;

reg [2:0] d;

wire max\_tick, min\_tick;

wire [2:0] q;

univ\_bin\_counter #(.N(3)) uut

(.clk(clk),

.reset(reset),

.syn\_clr(syn\_clr),

.load(load),

.en(en),

.up(up),

.d(d),

.max\_tick(max\_tick),

.min\_tick(min\_tick),

.q(q));

always

begin

clk = 1'b1;

#(T/2);

clk = 1'b0;

#(T/2);

end

initial

begin

reset = 1'b1;

#(T/2);

reset = 1'b0;

end

//Other stimulus

initial

begin

// === initial input ===

syn\_clr = 1'b0;

load = 1'b0;

en =1'b0;

up = 1'b1;

d = 3'b000;

@(negedge reset);

@(negedge clk);

load = 1'b1;

d = 3'b011;

@(negedge clk);

load = 1'b0;

repeat (2) @(negedge clk);

// === Test syn\_clr ====

syn\_clr = 1'b1;

@(negedge clk);

syn\_clr = 1'b0;

// == Test up counter and pause ==

en = 1'b1;

up = 1'b1;

repeat(10) @(negedge clk);

en =1'b0;

repeat(2) @(negedge clk);

en = 1'b1;

repeat(2) @(negedge clk);

// === test down counter ===

up = 1'b0;

repeat(10) @(negedge clk);

// === wait statement ===

// continue till q = 2

wait (q ==2);

@(negedge clk);univ\_bin\_counter #(.N(3)) uut

(.clk(clk),

.reset(reset),

.syn\_clr(syn\_clr),

.load(load),

.en(en),

.up(up),

.d(d),

.max\_tick(max\_tick),

.min\_tick(min\_tick),

.q(q));

always

begin

clk = 1'b1;

#(T/2);

clk = 1'b0;

#(T/2);

end

initial

begin

reset = 1'b1;

#(T/2);

reset = 1'b0;

end

//Other stimulus

initial

begin

// === initial input ===

syn\_clr = 1'b0;

load = 1'b0;

en =1'b0;

up = 1'b1;

d = 3'b000;

@(negedge reset);

@(negedge clk);

load = 1'b1;

d = 3'b011;

@(negedge clk);

load = 1'b0;

repeat (2) @(negedge clk);

// === Test syn\_clr ====

syn\_clr = 1'b1;

@(negedge clk);

syn\_clr = 1'b0;

// == Test up counter and pause ==

en = 1'b1;

up = 1'b1;

repeat(10) @(negedge clk);

en =1'b0;

repeat(2) @(negedge clk);

en = 1'b1;

repeat(2) @(negedge clk);

// === test down counter ===

up = 1'b0;

repeat(10) @(negedge clk);

// === wait statement ===

up = 1'b1;

// continue until min\_tick becomes 1

@(negedge clk);

wait(min\_tick);

@(negedge clk);

up = 1'b0;

// === absolute delay ===

#(4\*T);

en = 1'b0;

#(4\*T);

$stop;

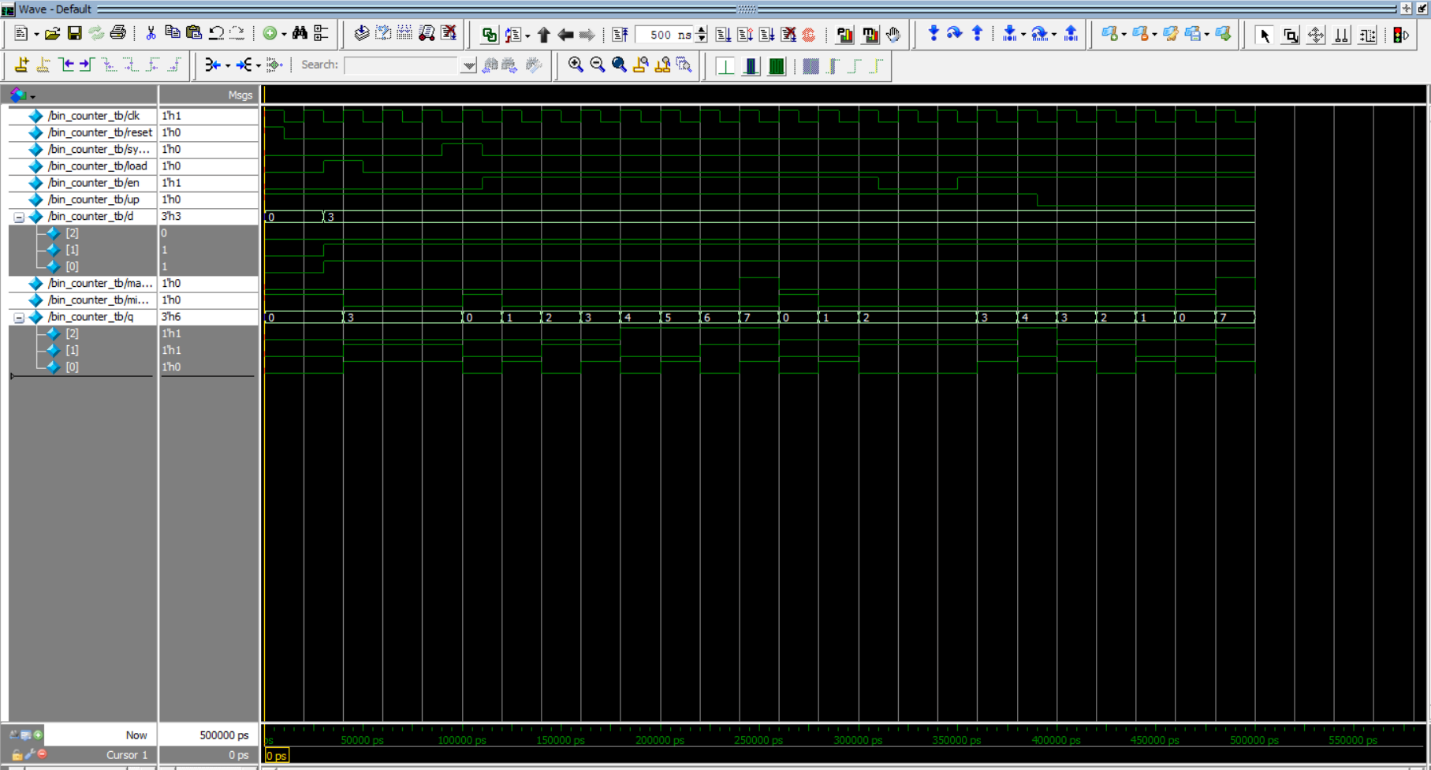
end

initial

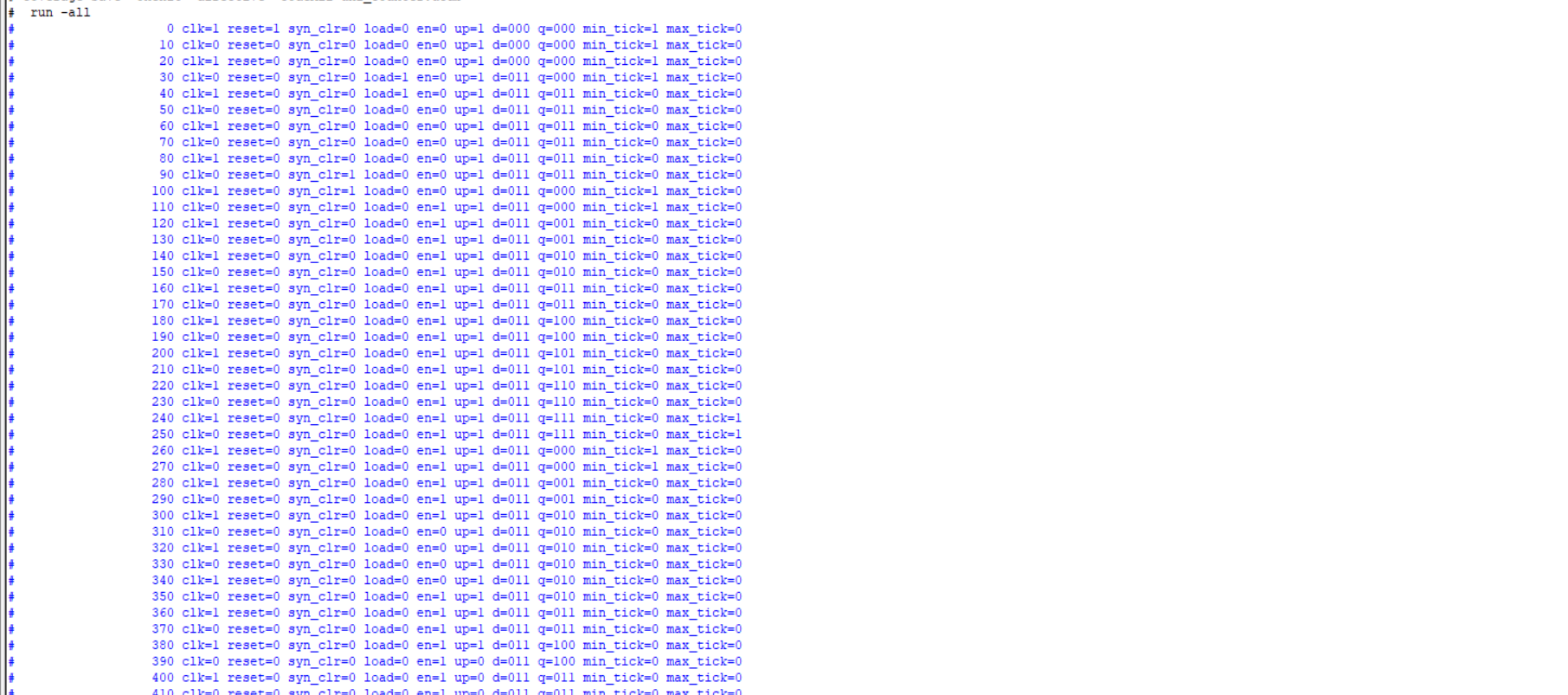
$monitor($time, " clk=%b reset=%b syn\_clr=%b load=%b en=%b up=%b d=%b q=%b min\_tick=%b max\_tick=%b", clk, reset,syn\_clr,load,en,up,d,q,min\_tick,max\_tick);

endmodule

**Wave\**



**Results**

****

**Web coverage report**

