HW 11 – Universal counter

Verilog

```
module univ_bin_counter
#(parameter N =8)
(
input wire clk,reset,
input wire syn_clr,load,en,up,
input wire [N-1:0] d,
output wire max_tick, min_tick,
output wire [N-1:0] q
);

// signal declaration
reg [N-1:0] r_reg, r_next;

//register
always @(posedge clk, posedge reset)
if(reset)
r_reg <= 0;
else r_reg <= r_next;

//next-state logic
always @*
if (syn_clr)
r_next = 0;
else if (load)
r_next = d;
else if (en & up)
r_next = r_reg + 1;
else if (en & ~up)
r_next = r_reg - 1;
else
r_next = r_reg;

// Output
assign q = r_reg;
assign max_tick = (r_reg==2**N-1) ? 1'b1 : 1'b0;
assign min_tick = (r_reg==0) ? 1'b1 : 1'b0;
endmodule
```

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
```

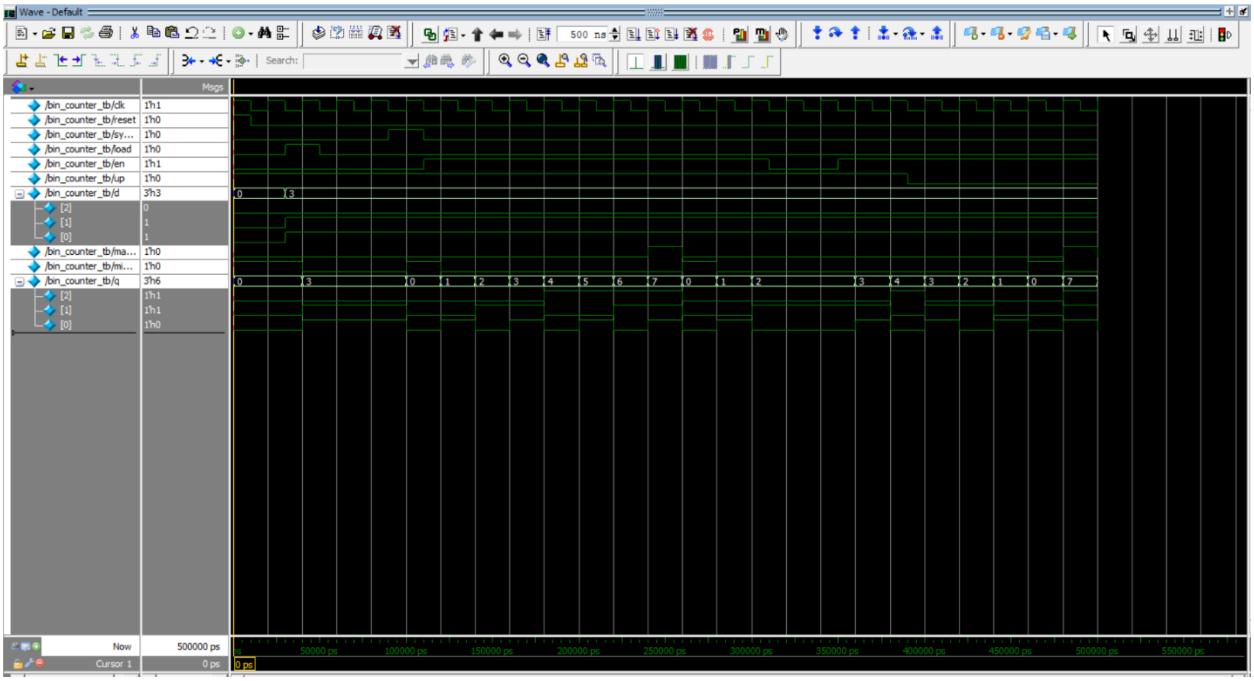
```
//Other stimulus
initial
begin
// === initial input ===
syn_clr = 1'b0;
load = 1'b0;
en =1'b0;
up = 1'b1;
d = 3'b000;
@(negedge reset);
@(negedge clk);

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
```

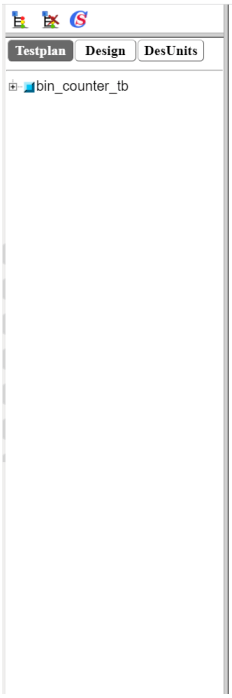
Wave



Results

```
# run -all
#                    0 clk=1 reset=1 syn_clr=0 load=0 en=0 up=1 d=000 q=000 min_tick=1 max_tick=0
#                   10 clk=0 reset=1 syn_clr=0 load=0 en=0 up=1 d=000 q=000 min_tick=1 max_tick=0
#                   20 clk=1 reset=0 syn_clr=0 load=0 en=0 up=1 d=000 q=000 min_tick=1 max_tick=0
#                   30 clk=0 reset=0 syn_clr=0 load=1 en=0 up=1 d=011 q=000 min_tick=1 max_tick=0
#                   40 clk=1 reset=0 syn_clr=0 load=1 en=0 up=1 d=011 q=011 min_tick=0 max_tick=0
#                   50 clk=0 reset=0 syn_clr=0 load=0 en=0 up=1 d=011 q=011 min_tick=0 max_tick=0
#                   60 clk=1 reset=0 syn_clr=0 load=0 en=0 up=1 d=011 q=011 min_tick=0 max_tick=0
#                   70 clk=0 reset=0 syn_clr=0 load=0 en=0 up=1 d=011 q=011 min_tick=0 max_tick=0
#                   80 clk=1 reset=0 syn_clr=0 load=0 en=0 up=1 d=011 q=011 min_tick=0 max_tick=0
#                   90 clk=0 reset=0 syn_clr=1 load=0 en=0 up=1 d=011 q=011 min_tick=0 max_tick=0
#                  100 clk=1 reset=0 syn_clr=1 load=0 en=0 up=1 d=011 q=000 min_tick=1 max_tick=0
#                  110 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=000 min_tick=1 max_tick=0
#                  120 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=001 min_tick=0 max_tick=0
#                  130 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=001 min_tick=0 max_tick=0
#                  140 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=010 min_tick=0 max_tick=0
#                  150 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=010 min_tick=0 max_tick=0
#                  160 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=011 min_tick=0 max_tick=0
#                  170 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=011 min_tick=0 max_tick=0
#                  180 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=100 min_tick=0 max_tick=0
#                  190 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=100 min_tick=0 max_tick=0
#                  200 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=101 min_tick=0 max_tick=0
#                  210 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=101 min_tick=0 max_tick=0
#                  220 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=110 min_tick=0 max_tick=0
#                  230 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=110 min_tick=0 max_tick=0
#                  240 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=111 min_tick=0 max_tick=1
#                  250 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=111 min_tick=0 max_tick=1
#                  260 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=000 min_tick=1 max_tick=0
#                  270 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=000 min_tick=1 max_tick=0
#                  280 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=001 min_tick=0 max_tick=0
#                  290 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=001 min_tick=0 max_tick=0
#                  300 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=010 min_tick=0 max_tick=0
#                  310 clk=0 reset=0 syn_clr=0 load=0 en=0 up=1 d=011 q=010 min_tick=0 max_tick=0
#                  320 clk=1 reset=0 syn_clr=0 load=0 en=0 up=1 d=011 q=010 min_tick=0 max_tick=0
#                  330 clk=0 reset=0 syn_clr=0 load=0 en=0 up=1 d=011 q=010 min_tick=0 max_tick=0
#                  340 clk=1 reset=0 syn_clr=0 load=0 en=0 up=1 d=011 q=010 min_tick=0 max_tick=0
#                  350 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=010 min_tick=0 max_tick=0
#                  360 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=011 min_tick=0 max_tick=0
#                  370 clk=0 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=011 min_tick=0 max_tick=0
#                  380 clk=1 reset=0 syn_clr=0 load=0 en=1 up=1 d=011 q=100 min_tick=0 max_tick=0
#                  390 clk=0 reset=0 syn_clr=0 load=0 en=1 up=0 d=011 q=100 min_tick=0 max_tick=0
#                  400 clk=1 reset=0 syn_clr=0 load=0 en=1 up=0 d=011 q=011 min_tick=0 max_tick=0
```

Web coverage report



ModelSim Coverage Report

Number of tests run:	1
Passed:	1
Warning:	0
Error:	0
Fatal:	0

List of tests included in report...

List of global attributes included in report...

List of Design Units included in report...

Coverage Summary by Structure:

Design Scope	Hits %	Coverage %
bin_counter_tb	91.53%	91.55%
uut	90.62%	92.50%

Coverage Summary by Type:

Total Coverage:					91.53%	91.55%
Coverage Type	**Bins**	**Hits**	**Misses**	**Weight**	**% Hit**	**Coverage**
Statements	59	59	0	1	100.00%	**100.00%**
Branches	7	7	0	1	100.00%	**100.00%**
FEC Expressions	2	2	0	1	100.00%	**100.00%**
FEC Conditions	4	3	1	1	75.00%	**75.00%**
Toggles	58	48	10	1	82.75%	**82.75%**

Report generated by ModelSim (ver. 10.7) on Monday 26 December 2022 10:23:13 with command line:
vcover report -html uni_counter.ucdb