**Load Counter Hot code**

module load\_cntr(

input wire clk,

input wire reset,

input wire load\_i,

input wire [3:0]load\_val\_i,

output wire [3:0]count\_o

);

logic[3:0] load\_ff;

always\_ff @(posedge clk or posedge reset)

if (reset)

load\_ff <= 4'h0;

else if (load\_i)

load\_ff <= load\_val\_i;

logic[3:0] count\_ff;

logic[3:0] nxt\_count;

always\_ff @(posedge clk or posedge reset)

if (reset)

count\_ff <= 4'h0;

else

count\_ff <= nxt\_count;

assign nxt\_count = load\_i ? load\_val\_i :

(count\_ff == 4'hF) ? load\_ff :

count\_ff + 4'h1;

assign count\_o = count\_ff;

endmodule

**Test Bench Code**

module tb\_load\_cntr();

logic clk;

logic reset;

logic load\_i;

logic[3:0] load\_val\_i;

logic[3:0] count\_o;

load\_cntr day10 (.\*);

always begin

clk = 1'b1;

#5;

clk = 1'b0;

#5;

end

int cycles;

initial begin

reset <= 1'b1;

load\_i <= 1'b0;

load\_val\_i <= 4'h0;

@(posedge clk);

reset <= 1'b0;

for (int i=0; i<3; i=i+1) begin

load\_i <= 1;

load\_val\_i <= 3\*i;

cycles = 4'hF - load\_val\_i[3:0];

@(posedge clk);

load\_i <= 0;

while (cycles) begin

cycles = cycles - 1;

@(posedge clk);

end

end

$finish();

end

initial begin

$dumpfile("day10.vcd");

$dumpvars(2, tb\_load\_cntr);

end

endmodule

**Simulation**

