`timescale 1 ms / 1 us

module TestBench ();

reg clk , reset ;

reg init , exp\_inc, exp\_dec;

reg NRE\_1 , NRE\_2 , ADC , expose , erase ;

// define the clock

always begin

clk = 1; #0.5; clk = 0; #0.5;

end

// Start pulses

initial begin

init = 0; reset = 1; exp\_inc = 0; exp\_dec = 0; clk = 1; #2;

reset = 0; #1;

init = 1; #1;

init = 0; #50;

//reset = 1; #1;

//reset = 0; #100;

$finish;

end

// Test camera with pulses

RE\_control dut(init , exp\_inc , exp\_dec, clk , reset , NRE\_1 , NRE\_2 , ADC , expose , erase );

endmodule

`timescale 1ms / 1us

module FSM\_ex\_control(input reg init, clk, reset, ovf5, output reg NRE\_1, NRE\_2, ADC, expose, erase, start);

parameter [1:0] idle = 2'b00;

parameter [1:0] exposure = 2'b01;

parameter [1:0] readout = 2'b10;

reg [1:0] state, next\_state;

reg [4:0] read\_counter;

always @(posedge clk) begin

if (reset) begin

state <= idle;

read\_counter <= 0;

end

else state <= next\_state;

if (state == readout) read\_counter <= read\_counter + 1;

end

always @ (\*) begin

case (state)

idle : begin

// Signaler som karakteristisk for state

erase <= 1;

expose <= 0;

NRE\_1 <= 1;

NRE\_2 <= 1;

ADC <= 0;

start <= 0;

// Init styrer hopp

if (init) begin

next\_state <= exposure;

start <= 1;

expose <= 0;

erase <= 1;

end

else next\_state <= state;

end

exposure : begin

erase <= 0;

start <= 0;

expose <= 1;

if (ovf5) begin

next\_state <= readout;

expose <= 0;

end

else next\_state <= exposure;

end

readout : begin

if (read\_counter == 0) NRE\_1 <= 0;

if (read\_counter == 1) ADC <= 1;

if (read\_counter == 2) ADC <= 0;

if (read\_counter == 3) NRE\_1 <= 1;

if (read\_counter == 4) NRE\_2 <= 0;

if (read\_counter == 5) ADC <= 1;

if (read\_counter == 6) ADC <= 0;

if (read\_counter == 7) NRE\_2 <= 1;

if (read\_counter == 8) begin

next\_state <= idle;

erase <= 1;

end

end

endcase

end

endmodule

`timescale 1 ms / 1 us

module CTRL\_ex\_time (input wire exp\_inc, exp\_dec, clk, reset, output reg [4:0] ex\_time);

always @(posedge clk) begin

if (reset) ex\_time <= 15;

if (exp\_inc & ex\_time<30) ex\_time <= ex\_time+1;

else if (exp\_dec & ex\_time>2) ex\_time <= ex\_time-1;

end

endmodule

`timescale 1 ms / 1 us

module Timer\_counter (input wire clk, reset, start, input reg [4:0] ex\_time, output reg ovf5);

reg [4:0] timer;

reg keep\_counting;

always @(posedge clk) begin

if (reset) begin

timer <= 0;

ovf5 <= 0;

keep\_counting = 0;

end

else if (start) begin

timer <= ex\_time - 1;

keep\_counting <= 1;

end

else if(timer != 0 && keep\_counting) timer <= timer -1;

else if (timer == 0 && keep\_counting) begin

ovf5 <= 1;

keep\_counting <= 0;

end

end

endmodule

`timescale 1 ms / 1 us

module RE\_control ( input reg init , exp\_inc, exp\_dec, clk , reset , output reg NRE\_1 , NRE\_2 , ADC , expose , erase );

reg ovf5 , start ;

reg [4:0] ex\_time ;

// Control EX\_time

CTRL\_ex\_time Ctrl\_ex ( exp\_inc, exp\_dec, clk , reset , ex\_time );

// Counter

Timer\_counter Timer (clk , reset , start, ex\_time , ovf5);

// FSM

FSM\_ex\_control FSM (init , clk , reset , ovf5, NRE\_1 , NRE\_2, ADC, expose , erase , start );

endmodule

`timescale 1 ms / 1 us

module TestBench ();

reg clk , reset ;

reg init , exp\_inc, exp\_dec;

reg NRE\_1 , NRE\_2 , ADC , expose , erase ;

// define the clock

always begin

clk = 1; #0.5; clk = 0; #0.5;

end

// Start pulses

initial begin

init = 0; reset = 1; exp\_inc = 0; exp\_dec = 0; clk = 1; #2;

reset = 0; #1;

init = 1; #1;

init = 0; #50;

//reset = 1; #1;

//reset = 0; #100;

$finish;

end

// Test camera with pulses

RE\_control dut(init , exp\_inc , exp\_dec, clk , reset , NRE\_1 , NRE\_2 , ADC , expose , erase );

endmodule