//Machine Lab project: DC motor control with FPGA cyclone 2 with variable duty cycle Pulse Width Modulation

//Lab group : 4

//Roll : 1606183, 1606182, 1606184, 1606185, 1606186, 1606187

module PWM\_Generator\_Verilog

(

clk, // 1KHz clock input from arduino mega

increase\_duty, // input to increase 10% duty cycle

decrease\_duty, // input to decrease 10% duty cycle

PWM\_OUT, //Output PWM signal

enable01, //Motor rotation control

enable02,

reverse // change the direction of rotation

);

input clk;

input reverse;

input increase\_duty;

input decrease\_duty;

output PWM\_OUT;

output enable01;

output enable02;

wire slow\_clk\_enable; // slow clock enable signal for debouncing FFs

reg[31:0] counter\_debounce=0;// counter for creating slow clock enable signals

wire tmp1,tmp2,duty\_inc;// temporary flip-flop signals for debouncing the increasing button

wire tmp3,tmp4,duty\_dec;// temporary flip-flop signals for debouncing the decreasing button

reg[27:0] counter\_PWM=0;// counter for creating 500Hz PWM signal

reg[31:0] DUTY\_CYCLE=5;

reg enable01 = 0;

reg enable02 = 1;// initial duty cycle is 50%

// Debouncing 2 buttons for inc/dec duty cycle

always @(posedge clk)

begin

counter\_debounce <= counter\_debounce + 1;

if(counter\_debounce>=2)

counter\_debounce <= 0;

end

assign slow\_clk\_enable = counter\_debounce == 2 ?1:0;

// debouncing FFs for increasing button

DFF\_PWM PWM\_DFF1(clk,slow\_clk\_enable,increase\_duty,tmp1);

DFF\_PWM PWM\_DFF2(clk,slow\_clk\_enable,tmp1, tmp2);

assign duty\_inc = tmp1 & (~ tmp2) & slow\_clk\_enable;

// debouncing FFs for decreasing button

DFF\_PWM PWM\_DFF3(clk,slow\_clk\_enable,decrease\_duty, tmp3);

DFF\_PWM PWM\_DFF4(clk,slow\_clk\_enable,tmp3, tmp4);

assign duty\_dec = tmp3 & (~ tmp4) & slow\_clk\_enable;

// vary the duty cycle using the debounced buttons above

always @(posedge clk)

begin

if(duty\_inc==1 && DUTY\_CYCLE <= 9)

DUTY\_CYCLE <= DUTY\_CYCLE + 1;// increase duty cycle by 10%

else if(duty\_dec==1 && DUTY\_CYCLE>=1)

DUTY\_CYCLE <= DUTY\_CYCLE - 1;//decrease duty cycle by 10%

end

// Create 500Hz PWM signal with variable duty cycle controlled by 2 buttons

always @(posedge clk)

begin

counter\_PWM <= counter\_PWM + 1;

if(counter\_PWM>=9)

counter\_PWM <= 0;

if (reverse) // reversing the input pins of motor driver

enable01 = ~enable01;

enable02 = ~enable02;

end

assign PWM\_OUT = counter\_PWM < DUTY\_CYCLE ? 1:0;

endmodule

// Debouncing DFFs for push buttons on FPGA

module DFF\_PWM(clk,en,D,Q);

input clk,en,D;

output reg Q;

always @(posedge clk)

begin

if(en==1) // slow clock enable signal

Q <= D;

end

endmodule