Course: CS 223 Digital Design

Lab: 04

Name: Yahya Mahmoud Ahmed Elnouby Mohamed

Section: 02

ID: 21801332

Date: 19/11/2019

Trainer: No. 19

System Verilog Code for the driver

```
`timescale 1ns / 1ps

module StepMotorState(
   input logic clk, reset, enable, direction, reverse,
   output logic A,Ab,B,Bb);

typedef enum logic [1:0] {S0, S1, S2, S3} State;
State currentState, nextState;

always_ff @(posedge clk)
if(reset) currentState <= S0;
else currentState <= nextState;

always_comb

case(currentState)
State0: if(enable && direction &&!reverse) nextState = State1;
   else if (enable && !direction && reverse) nextState = State3;
   else nextState = S0;

State1: if(enable && direction &&!reverse) nextState = State2;
   else if (enable && !direction && reverse) nextState = State0;
   else nextState = S1;

State2: if(enable && direction &&!reverse) nextState = State3;
   else if (enable && !direction && reverse) nextState = State1;
   else nextState = S2;

State3: if(enable && direction &&!reverse) nextState = State0;
   else if (enable && !direction && reverse) nextState = State2;
   else nextState = State3;

   default nextState = State0;

endcase

assign A = (currentState == State0 | currentState == State3);
assign Ab = (currentState == State0 | currentState == State1);
assign B = (currentState == State1 | currentState == State2);
assign Bb = (currentState == State2 | currentState == State3);

endmodule
```

TestBench Code for the driver

```
`timescale 1ns / 1ps

module StepMotorState_tb;

reg clk, reset, enable, direction, reverse;
wire A, Ab, B, Bb;

StepMotorState uut( .clk(clk), .reset(reset), .enable(enable),
.direction(direction), .reverse(reverse),
.A(A), .Ab(Ab),.B(B),.Bb(Bb) );
   initial
      begin
         reset <= 1; #10;
         reset <= 0; enable <= 1; direction <=1; reverse <=0; #10;
         enable <= 0; direction <= 1; reverse <=0; #15;
         enable <= 1; direction <=1;  reverse <=0; #10;
         enable <= 0; direction <= 1; reverse <=0; #10;
         enable <= 1; direction <=1;  reverse <=0; #10;
         enable <= 0; direction <= 1; reverse <=0; #10;
         enable <= 1; direction <=1;  reverse <=0; #10;
         enable <= 0; direction <= 1; reverse <=0; #10;
         enable <= 1; direction <=0;  reverse <=1; #10;
         enable <= 1; direction <=0;  reverse <=1; #10;
         enable <= 0; direction <= 0; reverse <=1; #30;
         enable <= 1; direction <=0;  reverse <=1; #10;
         enable <= 0; direction <= 0; reverse <=1; #10;
         enable <= 1; direction <=0;  reverse <=1; #10;
         enable <= 0; direction <= 0; reverse <=1; #10;
         enable <= 1; direction <=0;  reverse <=1; #10;
         enable <= 0; direction <= 0; reverse <=1; #20;

      end

   always
      begin
         clk <= 1; #5;
         clk <= 0; #5;
      end
endmodule
```

System Verilog for controlling the Speed using clock divider

```
`timescale 1ns / 1ps

module ClockDivider(input logic b0, b1, reset,
wire clk, output reg new_clk);

//counter
integer slowestCounter = 0;
integer mediumCounter = 0;
integer fastestCounter = 0;

//counter limits
localparam slowS = 15000000;
localparam mediumS = 10000000;
localparam fastS = 5000000

always@ (posedge clk)
begin
 if(reset)
   new_clk= 0;

   //The slowest speed for the car
   if (b0 && !b1)
   begin
       if(slowestCounter == slowS)
         begin
            new_clk <= ~new_clk;
            slowestCounter <=0;
         end
      else
         begin
            new_clk <= new_clk;
            slowestCounter <= slowestCounter+1;
         end
   end
```

```
//speed2 medium speed
   if (!b0 && b1)
   begin
      if(mediumCounter == mediumS)
         begin
            new_clk <= ~new_clk;
            mediumCounter <= 0;

         end
      else
         begin
            new_clk <= new_clk;
            mediumCounter <= mediumCounter+1;

         end
   end
    //speed3 fastest speed
    if (!b0 && !b1)
    begin

       if(fastestCounter == fastS)
         begin
         new_clk <= ~new_clk;
         fastestCounter <= 0;

         end
      else
         begin
            new_clk <= new_clk;
            fastestCounter <= fastestCounter+1;

         end
    end
end

endmodule
```

System Verilog Combining the state machine and speed (Code for driving a car)

```
`timescale 1ns / 1ps

module StepMotorInterface(input clk, enable, direction, reverse, reset, Speed0, Speed1,
 output A, Ab, B, Bb, a, b, c, d, e, f, g, dp,
 output [3:0] an );

//we will be used as the new clock after been divided
 wire new_clk;
 //this integers are to represent the decimal place in the seven segment display
 integer ones = 0;
 integer tens = 0;
 integer hundreds = 0;
 integer thousands = 0;

 //here is where the speed will be controlled according to the inputs
 always@(Speed0 or Speed1)
 begin
 if(Speed0 ==1 && Speed1 == 1) //the motor to stop for the given inputs
   begin
      ones <= 0;
      tens <= 0;
      hundreds <= 0;
      thousands <= 0;
   end

if(Speed0 ==1 && Speed1 == 0) //the motor to have low speed
   begin
      ones <= 0;
      tens <= 5;
      hundreds <= 2;
      thousands <= 0;
   end

if(Speed0 ==0 && Speed1 == 1) //the motor to have medium speed
   begin
      ones <= 0;
      tens <= 0;
      hundreds <= 5;
      thousands <= 0;
   end
```

```
if(Speed0 ==0 && Speed1 == 0) //the motor to have fast speed
   begin
      ones <= 0;
      tens <= 0;
      hundreds <= 0;
      thousands <= 1;
   end
end

 //in this part I am connecting the inputs in the interface using my FSM class done
 //to go through the states according to the inputs
 StepMotorState sm(.clk(new_clk), .reset(reset), .enable(enable), .direction(direction),
 .reverse(reverse), .A(A), .Ab(Ab), .B(B), .Bb(Bb));

 //This is for the clk divider in which we will use to control the speed the car
 ClockDivider div(.b0(Speed0), .b1(Speed1), .reset(reset), .clk(clk), .new_clk(new_clk));

 //this is to display the speed on the FPGA seven segment display
 SevSeg_4digit sevenDisplay(.clk(clk), .in0(ones), .in1(tens), .in2(hundreds),
.in3(thousands),
 .a(a), .b(b), .c(c), .d(d), .e(e), .f(f), .g(g), .dp(dp), .an(an));

endmodule
```

TestBench for simulating driving a car

```
`timescale 1ns / 1ps

module StepMotorInterface_tb;
reg clk, enable, direction,reverse, reset, Speed0, Speed1;
wire A, Ab, B, Bb, a, b, c, d, e, f, g, dp;
wire [3:0] an;

   StepMotorInterface uut( .clk(clk), .enable(enable), .direction(direction), .reverse(reverse),
    .reset(reset), .Speed0(Speed0), .Speed1(Speed1), .A(A), .Ab(Ab), .B(B), .Bb(Bb),
     .a(a), .b(b), .c(c), .d(d), .e(e), .f(f), .g(g), .dp(dp), .an(an));
   initial
   begin
      reset <= 0; #10;
      enable <= 1; direction <= 1; reverse <=0; Speed0 <= 0; Speed1 <= 0; #50;
      enable <= 1; direction <= 1; reverse <=0; Speed0 <= 0; Speed1 <= 1; #50;
      enable <= 1; direction <= 1; reverse <=0; Speed0 <= 1; Speed1 <= 0; #50;
      enable <= 1; direction <= 0; reverse <=1; Speed0 <= 0; Speed1 <= 0; #50;
      enable <= 1; direction <= 0; reverse <=1; Speed0 <= 0; Speed1 <= 1; #50;
      enable <= 1; direction <= 0; reverse <=1; Speed0 <= 1; Speed1 <= 0; #50;

   end

   always
      begin
         clk <= 1; #5;
         clk <= 0; #5;
      end

endmodule
```