```
/*
   Ben Davis
   2/1/24
   EE 371
   Lab 3, Task 2

   This module implements the line drawer, lab3_task1_inputter,
   and VGA_framebuffer modules onto an FPGA board. The used
   input of this module is switch 0.

*/

module DE1_SoC (HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, KEY, LEDR, SW, CLOCK_50,
   VGA_R, VGA_G, VGA_B, VGA_BLANK_N, VGA_CLK, VGA_HS, VGA_SYNC_N, VGA_VS);

   //fpga inputs and outputs
   output logic [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
   output logic [9:0] LEDR;
   input logic [3:0] KEY;
   input logic [9:0] SW;
   //clock signal
   input CLOCK_50;
   //vga screen outputs
   output [7:0] VGA_R;
   output [7:0] VGA_G;
   output [7:0] VGA_B;
   output VGA_BLANK_N;
   output VGA_CLK;
   output VGA_HS;
   output VGA_SYNC_N;
   output VGA_VS;

   //hex displays are off
   assign HEX0 = '1;
   assign HEX1 = '1;
   assign HEX2 = '1;
   assign HEX3 = '1;
   assign HEX4 = '1;
   assign HEX5 = '1;
   //switches light up if
   //corresponding switch is on
   assign LEDR = SW;

   //intermediate variables for
   //line_drawer and vga_framebuffer
   logic [9:0] x, a0, a1;
   logic [8:0] y, b0, b1;
   logic frame_start;
   logic pixel_color;

   logic ln_rst; //for resetting line_drawer independent of SW[0]

   //clock divider for slower animation
   logic [31:0] clk;
   clock_divider cdiv (.clock(CLOCK_50), .divided_clocks(clk));

   //animator module to update line
   animator t2 (.clk(clk[23]), .reset(~SW[0]), .x0(200), .x1(200), .w0(50), .w1(350),
                                       .a0(a0), .a1(a1));


   //////// DOUBLE_FRAME_BUFFER ////////
   logic dfb_en;
   assign dfb_en = 1'b0;
   /////////////////////////////////////

   VGA_framebuffer fb(.clk(CLOCK_50), .rst(~SW[0]), .x(x), .y(y),
            .pixel_color(1'b1), .pixel_write(SW[0]), .dfb_en, .frame_start,
            .VGA_R, .VGA_G, .VGA_B, .VGA_CLK, .VGA_HS, .VGA_VS,
            .VGA_BLANK_N, .VGA_SYNC_N);

   // draw lines between (x0, y0) and (x1, y1)
   line_drawer lines (.clk(CLOCK_50), .reset((~SW[0]) || (ln_rst)), .x0(a0), .y0(80),
                                 .x1(a1), .y1(80), .x(x), .y(y), .internal_rst(ln_rst));



endmodule

module DE1_SoC_testbench();

   //reset variables for testbench
   logic [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
   logic [9:0] LEDR;
   logic [3:0] KEY;
   logic [9:0] SW;
   logic CLOCK_50;
   logic [7:0] VGA_R;
   logic [7:0] VGA_G;
   logic [7:0] VGA_B;
   logic VGA_BLANK_N;
   logic VGA_CLK;
   logic VGA_HS;
   logic VGA_SYNC_N;
   logic VGA_VS;

   //clock setup
   parameter clk_pd = 100;
   initial begin
      CLOCK_50 <= 0;
      forever #(clk_pd /2) CLOCK_50 <= ~CLOCK_50;
   end //of clock setup

   //reinstantiate module
   DE1_SoC dut (.HEX0, .HEX1, .HEX2, .HEX3, .HEX4, .HEX5, .KEY, .LEDR, .SW, .CLOCK_50,
   .VGA_R, .VGA_G, .VGA_B, .VGA_BLANK_N, .VGA_CLK, .VGA_HS, .VGA_SYNC_N, .VGA_VS);

   //SW[4:1] are not used in this version anymore,
   //however SW[0] still is used as a "start" button,
   //tests to see if the system starts correctly.
   initial begin
      SW[4:0] <= 00000; @(posedge CLOCK_50 /16777216);
      SW[4:0] <= 00001; @(posedge CLOCK_50 /16777216);
      SW[4:0] <= 01001; @(posedge CLOCK_50 /16777216);
      SW[4:0] <= 10001; @(posedge CLOCK_50 /16777216);
      SW[4:0] <= 11001; @(posedge CLOCK_50 /16777216);
      $stop;
   end
endmodule //testbench end
```