```
/*
James Jhong
2/1/2022
EE371 Lab 3 Task 1

This module is the top level module for Task 1.
It calls the methods to create lines with Bresenham's line algroithm
and display the lines on a VGA display
*/
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);

   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;

   input CLOCK_50;
   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;

   assign HEX0 = '1;
   assign HEX1 = '1;
   assign HEX2 = '1;
   assign HEX3 = '1;
   assign HEX4 = '1;
   assign HEX5 = '1;
   assign LEDR = SW;

   logic [9:0] x0, x1, x;
   logic [8:0] y0, y1, y;
   logic frame_start;
   logic pixel_color;


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

   VGA_framebuffer fb(.clk(CLOCK_50), .rst(1'b0), .x, .y,
            .pixel_color, .pixel_write(1'b1), .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(1'b0),
            .x0, .y0, .x1, .y1, .x, .y);

//ex1
//    assign x0 = 50;
//    assign y0 = 0;
//    assign x1 = 50;
//    assign y1 = 240;
//    assign pixel_color = 1'b1;

//ex2
//    assign x0 = 0;
//    assign y0 = 240;
//    assign x1 = 240;
//    assign y1 = 240;
//    assign pixel_color = 1'b1;

//ex3
//    assign x0 = 0;
//    assign y0 = 50;
//    assign x1 = 240;
//    assign y1 = 0;
//    assign pixel_color = 1'b1;

```

```
77    //ex4
78    //    assign x0 = 240;
79    //    assign y0 = 50;
80    //    assign x1 = 0;
81    //    assign y1 = 0;
82    //    assign pixel_color = 1'b1;
83
84    //ex5
85    //    assign x0 = 0;
86    //    assign y0 = 0;
87    //    assign x1 = 120;
88    //    assign y1 = 240;
89    //    assign pixel_color = 1'b1;
90
91    //ex6
92    //    assign x0 = 0;
93    //    assign y0 = 240;
94    //    assign x1 = 50;
95    //    assign y1 = 0;
96    //    assign pixel_color = 1'b1;
97
98    //ex7
99       assign x0 = 37;
100      assign y0 = 47;
101      assign x1 = 573;
102      assign y1 = 350;
103      assign pixel_color = 1'b1;
104   endmodule
105
106   module DE1_SoC_testbench() ;
107      logic [3:0] KEY;
108      logic [9:0] SW;
109      logic [6:0] HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
110      logic [7:0] VGA_R;
111      logic [7:0] VGA_G;
112      logic [7:0] VGA_B;
113      logic VGA_BLANK_N;
114      logic VGA_CLK;
115      logic VGA_HS;
116      logic VGA_SYNC_N;
117      logic VGA_VS;
118      logic CLOCK_50;
119
120      DE1_SoC dut (HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, KEY, LEDR, SW, CLOCK_50,
121      VGA_R, VGA_G, VGA_B, VGA_BLANK_N, VGA_CLK, VGA_HS, VGA_SYNC_N, VGA_VS);
122
123      parameter CLOCK_PERIOD = 100;
124      initial begin
125         CLOCK_50 <= 0;
126         forever #(CLOCK_PERIOD/2) CLOCK_50 <= ~CLOCK_50; // Forever toggle the clock
127      end
128
129      initial begin
130            repeat(15) @(posedge CLOCK_50);
131         $stop;
132      end
133
134
135   endmodule
136
```