module control

(

input SetX,

input SetY,

input SetCol,

input GO,

input [8:0] sw,

input done,

output o\_x,

output o\_y,

output plot,

output color

);

enum int unsigned

{

S\_wait,

S\_start,

// S\_done // combined with start into one state

} state, nextstate;

logic signed [11:0] dx, dy, err, e2;  
logic right, down;

always\_ff @(posedge clk) begin

state <= S\_wait; // declare initial state  
 done <= 0;  
 plot <= 0;  
 // if (reset) state <= IDLE;  
 // else case (state)

case (state)  
S\_wait:  
 if (start) begin  
 dx = x1 - x0;  
 right = (dx >= 0);  
 if (~right) dx = -dx;  
 dy = y1 - y0;  
 down = dy >= 0;  
 if (down) dy = -dy;  
 err = dx + dy;  
 x <= x0;  
 y <= y0;  
 plot <= 1;  
 state <= S\_start;  
 end

S\_start:  
 if (x == x1 && y == y1) begin  
 done <= 1;  
 state <= S\_wait;  
 end else begin  
 plot <= 1;  
 e2 = err << 1;  
 if (e2 > dy) begin  
 err += dy;  
 if (right) x <= x + 9’d 1;  
 else x <= x - 9’d 1;  
 end  
 if (e2 < dx) begin  
 err += dx;  
 if (down) y <= y + 8’d 1;  
 else y <= y - 8’d 1;  
 end  
 end  
endcase  
end

endmodule