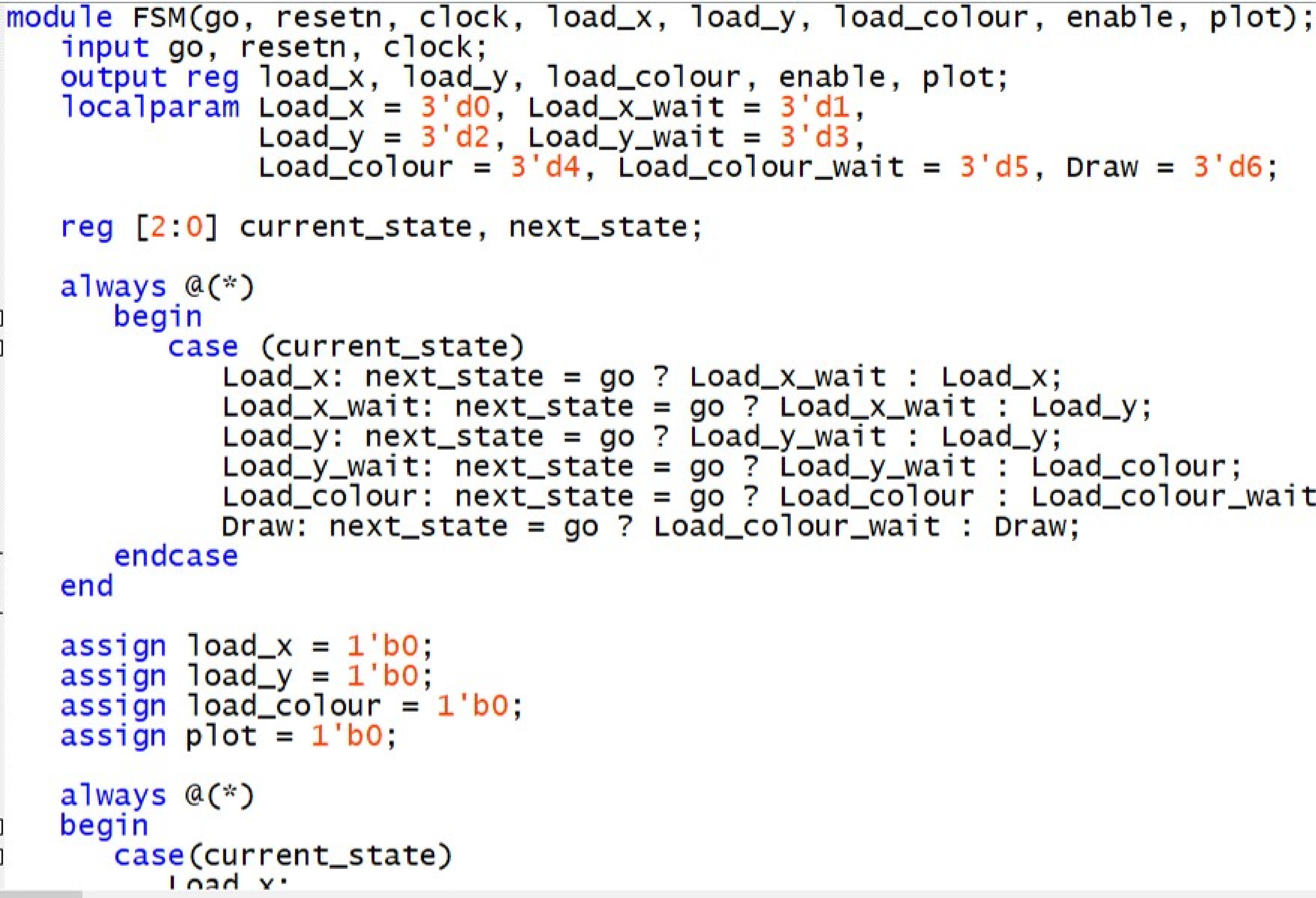

```
module FSM(go, resetn, clock, load_x, load_y, load_colour, enable, plot);
   input go, resetn, clock;
   output reg load_x, load_y, load_colour, enable, plot;
   localparam Load_x = 3'd0, Load_x_wait = 3'd1,
              Load_y = 3'd2, Load_y_wait = 3'd3,
              Load_colour = 3'd4, Load_colour_wait = 3'd5, Draw = 3'd6;

   reg [2:0] current_state, next_state;

   always @(*)
      begin
         case (current_state)
            Load_x: next_state = go ? Load_x_wait : Load_x;
            Load_x_wait: next_state = go ? Load_x_wait : Load_y;
            Load_y: next_state = go ? Load_y_wait : Load_y;
            Load_y_wait: next_state = go ? Load_y_wait : Load_colour;
            Load_colour: next_state = go ? Load_colour : Load_colour_wait
            Draw: next_state = go ? Load_colour_wait : Draw;
      endcase
   end

   assign load_x = 1'b0;
   assign load_y = 1'b0;
   assign load_colour = 1'b0;
   assign plot = 1'b0;

   always @(*)
   begin
      case(current_state)
         Load_x:
```