```
1   // Lab 5: Hazard Lights
2   // Michael Humphrey 2/6/2018 SID:1536326
3   module DE1_SoC (CLOCK_50, HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, KEY, LEDR, SW);
4      input  logic          CLOCK_50; // 50MHz clock.
5      output logic  [6:0]  HEX0, HEX1, HEX2, HEX3, HEX4, HEX5;
6      output logic  [9:0]  LEDR;
7      input  logic  [3:0]  KEY; // True when not pressed, False when pressed
8      input  logic  [9:0]  SW;
9
10     // Generate clk off of CLOCK_50, whichClock picks rate.
11     logic [31:0] clk;
12     parameter whichClock = 25;
13     clock_divider cdiv (CLOCK_50, clk);
14
15
16     // Hook up FSM inputs and outputs.
17
18     runway r (.clk(clk[whichClock]), .reset(~KEY[0]), .in(SW[1:0]), .out(LEDR[2:0]));
19
20     // Show signals on LEDRs so we can see what is happening.
21     assign LEDR[7] = clk[whichClock];
22     assign LEDR[5] = ~KEY[0];
23
24
25  endmodule
26
27  // divided_clocks[0] = 25MHz, [1] = 12.5Mhz, ... [23] = 3Hz, [24] = 1.5Hz, [25] = 0.75Hz,
    ...
28  module clock_divider (clock, divided_clocks);
29     input  logic          clock;
30     output logic  [31:0]  divided_clocks;
31
32     initial begin
33        divided_clocks <= 0;
34     end
35
36     always_ff @(posedge clock) begin
37        divided_clocks <= divided_clocks + 1;
38     end
39
40  endmodule
41
```