```
1   // Eugene Ngo
2   // 12/2/2022
3   // EE 271
4   // Lab 5
5
6   // clock_divider module generates 32 different clocks that can be used for slower clock
    speeds.
7
8   // Takes in a clock signal, divides the clock cycle and outputs 32
9   // divided clock signals of varying frequency.
10  // divided_clocks[0] = 25MHz, [1] = 12.5Mhz, ...
11  // [23] = 3Hz, [24] = 1.5Hz, [25] = 0.75Hz, ...
12  module clock_divider (clock, divided_clocks);
13
14     input logic clock;
15     output logic [31:0] divided_clocks = 32'b0;
16
17     always_ff @(posedge clock) begin
18        divided_clocks <= divided_clocks + 1;
19     end
20  endmodule
21
22  // clock_divider_testbench tests all expected, unexpected, and edgecase behaviors
23  module clock_divider_testbench();
24     logic clock;
25     logic [31:0] divided_clocks;
26
27     clock_divider dut (.clock, .divided_clocks);
28
29     parameter clock_period = 100;
30
31     integer i;
32     initial begin
33
34        for (i = 0; i < clock_period; i++) begin
35
36           @(posedge clock);
37
38        end // for loop
39
40        $stop; // ends the simulation
41
42     end // initial
43  endmodule
```