```
1   /*
2      Ben Davis
3      1/23/24
4      EE 371
5      Lab 2, Task 2
6
7      This module takes a clock period (frequency in Hz) and continuously
8      divides each frequency by 1/2 until it reaches its 32nd index
9
10  */
11
12  module clock_divider (clock, divided_clocks);
13
14     input logic clock;
15     //output array of 32 frequencies
16     output logic [31:0] divided_clocks = 32'b0;
17
18     always_ff @(posedge clock) begin
19        divided_clocks <= divided_clocks + 1; //slows down clock
20     end
21  endmodule
22
23
24  module clock_divider_testbench();
25     logic clock;
26     logic [31:0] divided_clocks;
27
28     clock_divider dut (.clock, .divided_clocks);
29
30
31     parameter clock_period = 100;
32
33     initial begin
34
35        clock <= 0;
36        forever #(clock_period /2) clock <= ~clock; //cut frequency in half
37     end //end initialization
38
39     initial begin
40
41          for (int i = 0; i < 100; i++) begin
42              @(posedge clock); //loop through
43          end
44
45          $stop; // Stop the simulation
46      end
47  endmodule
48
```