```
1   // Eugene Ngo
2   // 4/7/2023
3   // CSE 469
4   // Lab 1 Task 2
5
6   // reg_file__testbench tests all expected, unexpected, and edgecase behaviors
7   module reg_file_testbench();
8      logic clk, wr_en;
9      logic [31:0] write_data, read_data1, read_data2;
10     logic [3:0] write_addr, read_addr1, read_addr2;
11
12     reg_file dut (.clk, .wr_en, .write_data, .write_addr, .read_addr1, .read_addr2, .
    read_data1, .read_data2);
13
14     parameter clock_period = 10000;
15
16     integer i;
17
18     initial begin // Set up the clock
19        clk <= 0;
20        for (i=0; i<1000; i++) begin: clockCount
21           forever #(clock_period /2) clk <= ~clk;
22        end
23
24     end
25
26     initial begin
27        $display("%t Behavior check", $time);
28
29              // Testing functionality of
30              // 1 cycle delay of writes.
31        wr_en = 1'b1;                  @(posedge clk);
32        write_data = 32'b0;            @(posedge clk);
33        write_addr = 4'b0010;          @(posedge clk);
34                                       @(posedge clk);
35
36        write_data = 32'b1;            @(posedge clk);
37        write_addr = 4'b0011;          @(posedge clk);
38                                       @(posedge clk);
39              // Testing functionality of
40              // same cycle reads.
41        read_addr1 = 4'b0010;          @(posedge clk);
42        read_addr2 = 4'b0011;          @(posedge clk);
43
44              // Testing the functionality of
45              // 1 cycle delayed reads
46              // after an updated write
47        write_addr = 4'b0010;          @(posedge clk);
48        read_addr1 = 4'b0010;          @(posedge clk);
49                                       @(posedge clk);
50        // read_data1 should update to 1 now.
51
52        write_data = 32'b0;            @(posedge clk);
53        write_addr = 4'b0011;          @(posedge clk);
54
55        read_addr2 = 4'b0011;          @(posedge clk);
56                                       @(posedge clk);
57        // read_data2 should update to 0 now.
58
59
60
61
62     end
63
64
65
66
67  endmodule
68
```