```
1   //Aaron Hong (ahong02)
2   //Stephen Macris (smacris)
3   //3/29/23
4   //EE469 Lab1
5
6   //This testbench tests reg_file to ensure that it functions properly: mostly testing
    asynchronous read.
7
8   module reg_file_testbench();
9
10        logic clk, wr_en;
11        logic [31:0] write_data, read_data1, read_data2;
12        logic [3:0] write_addr, read_addr1, read_addr2;
13
14        reg_file dut (.clk(clk), .wr_en(wr_en), .write_data(write_data), .write_addr(
    write_addr), .read_addr1(read_addr1), .read_addr2(read_addr2), .read_data1(read_data1), .
    read_data2(read_data2));
15
16        //clock setup
17        parameter clock_period = 100;
18        initial begin
19           clk <= 0;
20           forever #(clock_period /2) clk <= ~clk;
21
22        end //initial
23
24        initial begin
25
26           wr_en <= 1'b0; write_addr <= 4'b0010; write_data <= 32'd8; read_addr1 <= 4'b0000;
    read_addr2 <= 4'b0001;  @(posedge clk);
27           wr_en <= 1'b1;

    @(posedge clk);
28           wr_en <= 1'b0; write_addr <= 4'b0100; write_data <= 32'd16; read_addr1 <= 4'b0000;
    read_addr2 <= 4'b0001; @(posedge clk);
29           wr_en <= 1'b1;

    @(posedge clk);
30
                          @(posedge clk);
31           wr_en <= 1'b0; read_addr1 <= 4'b0010;
                                                                    @(posedge clk);
32           wr_en <= 1'b0; read_addr2 <= 4'b0100;
                                                                    @(posedge clk);
33           wr_en <= 1'b1; write_addr <= 4'b0011; write_data <= 32'd32; read_addr2 <= 4'b0011;
                            @(posedge clk);
34
                          @(posedge clk);
35           wr_en <= 1'b0;

    @(posedge clk);
36           wr_en <= 1'b1; write_addr <= 4'b0111; write_data <= 32'd64; read_addr1 <= 4'b0111;
                            @(posedge clk);
37
                          @(posedge clk);
38
                          @(posedge clk);
39
                          @(posedge clk);
40
41           $stop; //end simulation
42
43        end //initial
44
45  endmodule
46
```