```
1   // Eugene Ngo
2   // 5/3/23
3   // EE 469
4   // Lab 3
5
6   // reg_file_testbench is a testing fiel for reg_file
7   // that tests  that write data is written
8   // into register file a cycle after wr_en is true,
9   // checks if read data updates the register data
10  // the same cycle as the address asserted,
11  // and checks if read data is updated to write data
12  // the cycle after address is provided if
13  // write address is the same and wr_en is true
14  module reg_file_testbench();
15     logic CLOCK_50;
16     logic clk;
17     logic wr_en;
18     logic [31:0] write_data;
19     logic [3:0]  write_addr;
20     logic [3:0]  read_addr1, read_addr2;
21     logic [31:0] read_data1, read_data2;
22     assign CLOCK_50 = clk;
23
24     reg_file dut (.clk, .wr_en, .write_data, .write_addr, .read_addr1,. read_addr2, .
       read_data1, .read_data2);
25     parameter CLOCK_PERIOD=100;
26     initial begin
27        clk <= 0;
28        forever #(CLOCK_PERIOD/2) clk <= ~clk;
29     end
30
31
32     initial begin
33        wr_en <= 0; write_data = 5; write_addr = 3; read_addr1 = 2; read_addr2 = 3; @(posedge
       clk);
34        wr_en <= 1;  repeat(1) @(posedge clk);
35        write_data = 10; write_addr = 4;@(posedge clk);
36        write_data = 11; write_addr = 5;@(posedge clk);
37        read_addr1 = 4; @(posedge clk);
38        read_addr2 = 5; @(posedge clk);
39        write_data = 12; write_addr = 4; @(posedge clk);
40        write_data = 13; write_addr = 5; @(posedge clk);
41
42     end
43  endmodule
```