```
// Eugene Ngo
// 4/7/2023
// CSE 469
// Lab 1 Task 2

// reg_file__testbench tests all expected, unexpected, and edgecase behaviors
module reg_file_testbench();
   logic clk, wr_en;
   logic [31:0] write_data, read_data1, read_data2;
   logic [3:0] write_addr, read_addr1, read_addr2;

   reg_file dut (.clk, .wr_en, .write_data, .write_addr, .read_addr1, .read_addr2, .read_data1, .read_data2);

   parameter clock_period = 10000;

   integer i;

   initial begin // Set up the clock
      clk <= 0;
      for (i=0; i<1000; i++) begin: clockCount
         forever #(clock_period /2) clk <= ~clk;
      end

   end

   initial begin
      $display("%t Behavior check", $time);

            // Testing functionality of
            // 1 cycle delay of writes.
      wr_en = 1'b1;                 @(posedge clk);
      write_data = 32'b0;           @(posedge clk);
      write_addr = 4'b0010;         @(posedge clk);
                                    @(posedge clk);

      write_data = 32'b1;           @(posedge clk);
      write_addr = 4'b0011;         @(posedge clk);
                                    @(posedge clk);
            // Testing functionality of
            // same cycle reads.
      read_addr1 = 4'b0010;         @(posedge clk);
      read_addr2 = 4'b0011;         @(posedge clk);

            // Testing the functionality of
            // 1 cycle delayed reads
            // after an updated write
      write_addr = 4'b0010;         @(posedge clk);
      read_addr1 = 4'b0010;         @(posedge clk);
                                    @(posedge clk);
      // read_data1 should update to 1 now.

      write_data = 32'b0;           @(posedge clk);
      write_addr = 4'b0011;         @(posedge clk);

      read_addr2 = 4'b0011;         @(posedge clk);
                                    @(posedge clk);
      // read_data2 should update to 0 now.




   end




endmodule

```