```
//Aaron Hong (ahong02)
//Stephen Macris (smacris)
//3/29/23
//EE469 Lab1

//This module creates an asynchronous, two read port, one write port, 16x32 register file.

//Inputs: Two 1-bits clk (clock signal), wr_en (write enable), three 4-bits write_addr, read_addr1,
//read_addr2 (write and read addresses), one 32-bit write_data (data to be written at given write address).
//Outputs: Two 32-bits read_data1, read_data2 (data read from given read addresses).
module reg_file (clk, wr_en, write_data, write_addr, read_addr1, read_addr2, read_data1, read_data2);

   input  logic  clk, wr_en;
   input  logic [3:0] write_addr, read_addr1, read_addr2;
   input  logic [31:0] write_data;
   output logic [31:0] read_data1, read_data2;

   logic [15:0][31:0] memory;

   always_ff @(posedge clk) begin
      if(wr_en) begin
         memory[write_addr] <= write_data;
      end
   end

   always_comb begin
      read_data1 <= memory[read_addr1];
      read_data2 <= memory[read_addr2];
   end

endmodule
```